OPTIMIZACIÓN DE LA EJECUCIÓN DE FLUJOS DE TRABAJOS EMPRESARIALES EN INFRAESTRUCTURAS CLOUD

RICARDO FRANCISCO MENDOZA DIEZ

MÁSTER EN INGENIERÍA INFORMÁTICA, FACULTAD DE INFORMÁTICA,

UNIVERSIDAD COMPLUTENSE DE MADRID

Trabajo Fin Máster en Ingeniería Informática

Septiembre, 2016

Curso académico: 2015/2016

Calificación obtenida: 6,7

Director:

José Luis Vázquez-Poletti

Autorización de difusión y utilización

El abajo firmante, matriculado en el Máster en Ingeniería Informática de la Facultad de Informática, autoriza a la Universidad Complutense de Madrid (UCM) a difundir y utilizar con fines académicos, no comerciales y mencionando expresamente a su autor el presente Trabajo Fin de Máster: "Optimización de la ejecución de flujos de trabajos empresariales en infraestructuras Cloud", realizado durante el curso académico 2015-2016, bajo la dirección de José Luis Vázquez-Poletti que pertenece al Departamento de Arquitectura de Computadores y Automática, y a la Biblioteca de la UCM a depositarlo en el Archivo Institucional E-Prints Complutense con el objeto de incrementar la difusión, uso e impacto del trabajo en internet y garantizar su preservación y acceso a largo plazo.

Ricardo Francisco Mendoza Diez

Madrid, 20 de Setiembre de 2016

Agradecimientos

A Dios, por haberme permitido llegar hasta este punto y haberme dado mucha salud para lograr mis objetivos.

A mi director José Luis Vázquez-Poletti, por darme la oportunidad de desarrollar este trabajo, por su disponibilidad en todo momento para ayudarme a resolver cualquier duda y por los conocimientos transmitidos durante el desarrollo de este proyecto.

Al Ministerio de Educación del Perú, por otorgarme una beca para poder estudiar el máster.

A mis padres, a mi hermana y a mi novia Lucy, por todo el amor y apoyo incondicional que me dan todos los días.

A mis amigos y compañeros de la universidad, en especial a Steph por su apoyo y sus consejos.

Resumen

En la actualidad, el uso del Cloud Computing se está incrementando y existen muchos proveedores que ofrecen servicios que hacen uso de esta tecnología. Uno de ellos es Amazon Web Services, que a través de su servicio Amazon EC2, nos ofrece diferentes tipos de instancias que podemos utilizar según nuestras necesidades. El modelo de negocio de AWS se basa en el pago por uso, es decir, solo realizamos el pago por el tiempo que se utilicen las instancias.

En este trabajo se implementa en Amazon EC2, una aplicación cuyo objetivo es extraer de diferentes fuentes de información, los datos de las ventas realizadas por las editoriales y librerías de España. Estos datos son procesados, cargados en una base de datos y con ellos se generan reportes estadísticos, que ayudarán a los clientes a tomar mejores decisiones.

Debido a que la aplicación procesa una gran cantidad de datos, se propone el desarrollo y validación de un modelo, que nos permita obtener una ejecución óptima en Amazon EC2. En este modelo se tienen en cuenta el tiempo de ejecución, el coste por uso y una métrica de coste/rendimiento.

Adicionalmente, se utilizará la tecnología de contenedores Docker para llevar a cabo un caso específico del despliegue de la aplicación.

Palabras claves: Cloud Computing, Amazon EC2, aplicación ETL, Docker, instancia, coste, rendimiento, tiempo de ejecución, contenedor.

Abstract

Currently, the use of cloud computing is increasing and there are many providers offering services using this technology. One of them is Amazon Web Services and through its Amazon EC2 service, offers different types of instances that can be used according to our needs. The AWS business model is based on pay per use, so we only pay for the time that instances are used.

In this paper, has been implemented in Amazon EC2, an application whose goal is to extract information from different sources, data from sales by publishers and bookstores in Spain. These data are processed, loaded into a database and with them generate statistical reports, which will help customers make better decisions are generated.

Because the application processes a large amount of data, the development and validation of a model that allows us to achieve optimum performance in Amazon EC2 is proposed. This model considers the runtime, cost per use and a metric of cost / performance.

Additionally, the Docker container technology will be used to carry out a specific case of application deployment.

Key words: Cloud Computing, Amazon EC2, ETL application, Docker, instance, cost, performance, runtime, container.

Lista de acrónimos

SaaS	Software as a Service
PaaS	Platform as a Service
IaaS	Infrastructure as a Service
AWS	Amazon Web Services
EC2	Elastic Compute Cloud
IOPS	Operaciones de entrada/salida por segundo
SSD	Unidad de estado sólido
E/S	Entrada/Salida
TB	Terabyte
ETL	Extraer, transformar y cargar
GiB	Gibibyte
vCPU	CPU virtual
TI	Tecnología de Información
C/R	Coste/Rendimiento

Índice general

Resumen ..... **II**

Abstract ..... **III**

Lista de acrónimos ..... **IV**

Índice de Figuras ..... **VII**

1. Introducción **1**

1.1. Objetivo de la investigación 2

1.2. Estructura del trabajo 2

1. Introduction **4**

1.1. Objective research 5

1.2. Document Structure 5

2. Cloud Computing **6**

2.1. Tipos de servicios 6

2.2. Modelos de despliegue 7

2.3. Beneficios y riesgos 8

3. Amazon Web Services **10**

3.1. Amazon Elastic Compute Cloud (Amazon EC2) 11

3.1.1. Tipos de Instancia EC2 13

4. Docker **14**

4.1. Contenedor y Máquina virtual 14

4.2. Arquitectura de Docker 16

4.3. Contenedores como Servicio (CaaS) 17

4.4. Funcionamiento de Docker 18

4.5. Herramientas de Docker 19

5. Aplicación ETL **20**

5.1. Aplicación ETL en Amazon EC2 21

5.2. Aplicación ETL en Amazon EC2 con Docker 22

6. Modelo de Ejecución **23**

6.1. Resultados experimentales 23

6.1.1. En Amazon EC2 23

6.1.2. En Amazon EC2 con Docker 26
6.2. Fórmula del Modelo 29

7. Resultados Analíticos **31**
7.1. En Amazon EC2 31
7.1.1. Tiempo 31
7.1.2. Coste/Rendimiento 36
7.2. En Amazon EC2 con Docker 40
7.2.1. Tiempo 41
7.2.2. Coste/Rendimiento 45

8. Casos de Uso **51**

9. Conclusiones y Trabajo futuro **59**

9. Conclusions and Future Work **60**

Referencias Bibliográficas **61**

Índice de Figuras

Figura 3.1. Servicios que ofrece Amazon Web Services. 10

Figura 3.2. Infraestructura global de Amazon Web Services..... 12

Figura 4.1. Enfoque arquitectónico de (a) máquina virtual y (b) contenedor Docker..... 15

Figura 4.2. Arquitectura de Docker..... 16

Figura 4.3. Esquema del funcionamiento de Docker. 18

Figura 5.1. Esquema de la Aplicación ETL. 20

Figura 6.1. Tiempos de ejecución en la región EU (Frankfurt) de Amazon EC2. 23

Figura 6.2. Tiempos de ejecución en la región EEUU Este (Norte de Virginia) de Amazon EC2. 24

Figura 6.3. Tiempos de ejecución en local y en la región EU (Frankfurt) de Amazon EC2..... 25

Figura 6.4. Tiempos de ejecución en local y en la región EEUU Este (Norte de Virginia) de Amazon EC2. 25

Figura 6.5. Tiempos de ejecución en la región EU (Frankfurt) de Amazon EC2. 26

Figura 6.6. Tiempos de ejecución en la región EEUU Este (Norte de Virginia) de Amazon EC2. 27

Figura 6.7. Tiempos de ejecución en local y en la región EU (Frankfurt) de Amazon EC2 con Docker. 28

Figura 6.8. Tiempos de ejecución en local y en la región EEUU Este (Norte de Virginia) de Amazon EC2 con Docker..... 28

Figura 7.1. Tiempos al procesar 1 millón de ficheros en la región EU (Frankfurt) de Amazon EC2..... 31

Figura 7.2. Tiempos al procesar 2 millones de ficheros en la región EU (Frankfurt) de Amazon EC2..... 32

Figura 7.3. Tiempos al procesar 3 millones de ficheros en la región EU (Frankfurt) de Amazon EC2..... 32

Figura 7.4. Tiempos al procesar 4 millones de ficheros en la región EU (Frankfurt) de Amazon EC2..... 32

Figura 7.5. Tiempos al procesar 5 millones de ficheros en la región EU (Frankfurt) de Amazon EC2..... 33

Figura 7.6. Tiempos al procesar 6 millones de ficheros en la región EU (Frankfurt) de Amazon EC2..... 33

Figura 7.7. Tiempos al procesar 1 millón de ficheros en región EE.UU (Norte de Virginia) de Amazon EC2. 33

Figura 7.8. Tiempos al procesar 2 millones de ficheros en la región EE.UU (Norte de Virginia) de Amazon EC2. 34

Figura 7.9. Tiempos al procesar 3 millones de ficheros en la región EE.UU (Norte de Virginia) de Amazon EC2. 34

Figura 7.10. Tiempos al procesar 4 millones de ficheros en la región EE.UU (Norte de Virginia) de Amazon EC2. 34

Figura 7.11. Tiempos al procesar 5 millones de ficheros en la región EE.UU (Norte de Virginia) de Amazon EC2. 35

Figura 7.12. Tiempos al procesar 6 millones de ficheros en la región EE.UU (Norte de Virginia) de Amazon EC2. 35

Figura 7.13. Valores C/R al procesar 1 millón de ficheros en la región EU (Frankfurt) de Amazon EC2. 36

Figura 7.14. Valores C/R al procesar 2 millones de ficheros en la región EU (Frankfurt) de Amazon EC2. 36

Figura 7.15. Valores C/R al procesar 3 millones de ficheros en la región EU (Frankfurt) de Amazon EC2. 37

Figura 7.16. Valores C/R al procesar 4 millones de ficheros en la región EU (Frankfurt) de Amazon EC2. 37

Figura 7.17. Valores C/R al procesar 5 millones de ficheros en la región EU (Frankfurt) de Amazon EC2. 37

Figura 7.18. Valores C/R al procesar 6 millones de ficheros en la región EU (Frankfurt) de Amazon EC2. 38

Figura 7.19. Valores C/R al procesar 1 millón de ficheros en la región EE.UU (Norte de Virginia) de Amazon EC2. 38

Figura 7.20. Valores C/R al procesar 2 millones de ficheros en la región EE.UU (Norte de Virginia) de Amazon EC2. 38

Figura 7.21. Valores C/R al procesar 3 millones de ficheros en la región EE.UU (Norte de Virginia) de Amazon EC2. 39

Figura 7.22. Valores C/R al procesar 4 millones de ficheros en la región EE.UU (Norte de Virginia) de Amazon EC2. 39

Figura 7.23. Valores C/R al procesar 5 millones de ficheros en la región EE.UU (Norte de Virginia) de Amazon EC2. 39

Figura 7.24. Valores C/R al procesar 6 millones de ficheros en la región EE.UU (Norte de Virginia) de Amazon EC2. 40

Figura 7.25. Tiempos al procesar 1 millón de ficheros en la región EU (Frankfurt) de Amazon EC2 con Docker. 41

Figura 7.26. Tiempos al procesar 2 millones de ficheros en la región EU (Frankfurt) de Amazon EC2 con Docker. 41

Figura 7.27. Tiempos al procesar 3 millones de ficheros en la región EU (Frankfurt) de Amazon EC2 con Docker. 42

Figura 7.28. Tiempos al procesar 4 millones de ficheros en la región EU (Frankfurt) de Amazon EC2 con Docker. 42

Figura 7.29. Tiempos al procesar 5 millones de ficheros en la región EU (Frankfurt) de Amazon EC2 con Docker..... 42

Figura 7.30. Tiempos al procesar 6 millones de ficheros en la región EU (Frankfurt) de Amazon EC2 con Docker..... 43

Figura 7.31. Tiempos al procesar 1 millón de ficheros en la región EE.UU (Norte de Virginia) de Amazon EC2 con Docker..... 43

Figura 7.32. Tiempos al procesar 2 millones de ficheros en la región EE.UU (Norte de Virginia) de Amazon EC2 con Docker. 43

Figura 7.33. Tiempos al procesar 3 millones de ficheros en la región EE.UU (Norte de Virginia) de Amazon EC2 con Docker. 44

Figura 7.34. Tiempos al procesar 4 millones de ficheros en la región EE.UU (Norte de Virginia) de Amazon EC2 con Docker. 44

Figura 7.35. Tiempos al procesar 5 millones de ficheros en la región EE.UU (Norte de Virginia) de Amazon EC2 con Docker. 44

Figura 7.36. Tiempos al procesar 6 millones de ficheros en la región EE.UU (Norte de Virginia) de Amazon EC2 con Docker. 45

Figura 7.37. Valores C/R al procesar 1 millón de ficheros en la región EU (Frankfurt) de Amazon EC2 con Docker..... 46

Figura 7.38. Valores C/R al procesar 2 millones de ficheros en la región EU (Frankfurt) de Amazon EC2 con Docker..... 46

Figura 7.39. Valores C/R al procesar 3 millones de ficheros en la región EU (Frankfurt) de Amazon EC2 con Docker..... 46

Figura 7.40. Valores C/R al procesar 4 millones de ficheros en la región EU (Frankfurt) de Amazon EC2 con Docker..... 47

Figura 7.41. Valores C/R al procesar 5 millones de ficheros en la región EU (Frankfurt) de Amazon EC2 con Docker..... 47

Figura 7.42. Valores C/R al procesar 6 millones de ficheros en la región EU (Frankfurt) de Amazon EC2 con Docker..... 47

Figura 7.43. Valores C/R al procesar 1 millón de ficheros en la región EE.UU (Norte de Virginia) de Amazon EC2 con Docker. 48

Figura 7.44. Valores C/R al procesar 2 millones de ficheros en la región EE.UU (Norte de Virginia) de Amazon EC2 con Docker. 48

Figura 7.45. Valores C/R al procesar 3 millones de ficheros en la región EE.UU (Norte de Virginia) de Amazon EC2 con Docker. 48

Figura 7.46. Valores C/R al procesar 4 millones de ficheros en la región EE.UU (Norte de Virginia) de Amazon EC2 con Docker. 49

Figura 7.47. Valores C/R al procesar 5 millones de ficheros en la región EE.UU (Norte de Virginia) de Amazon EC2 con Docker. 49

Figura 7.48. Valores C/R al procesar 6 millones de ficheros en la región EE.UU (Norte de Virginia) de Amazon EC2 con Docker. 49

Figura 8.1. Valores C/R al procesar 2 millones de ficheros, (a) en la Región EU (Frankfurt) y (b) en la Región EE.UU. (Norte de Virginia) de Amazon EC2. 51

Figura 8.2. Valores C/R al procesar 2 millones de ficheros, (a) en la Región EU (Frankfurt) y (b) en la Región EE.UU. (Norte de Virginia) de Amazon EC2 con Docker..... 52

Figura 8.3. Valores C/R al procesar 4 millones de ficheros, (a) en la Región EU (Frankfurt) y (b) en la Región EE.UU. (Norte de Virginia) de Amazon EC2. 53

Figura 8.4. Valores C/R al procesar 4 millones de ficheros, (a) en la Región EU (Frankfurt) y (b) en la Región EE.UU. (Norte de Virginia) de Amazon EC2 con Docker..... 54

Figura 8.5. Valores C/R al procesar 5 millones de ficheros y utilizando 20 instancias, (a) en la Región EU (Frankfurt) y (b) en la Región EE.UU. (Norte de Virginia) de Amazon EC2..... 55

Figura 8.6. Valores C/R al procesar 5 millones de ficheros y utilizando 20 instancias, (a) en la Región EU (Frankfurt) y (b) en la Región EE.UU. (Norte de Virginia) de Amazon EC2 con Docker. 56

Figura 8.7. Valores C/R al procesar 6 millones de ficheros, utilizando 10 instancias y en un máximo de 3 horas, (a) en la Región EU (Frankfurt) y (b) en la Región EE.UU. (Norte de Virginia) de Amazon EC2. 57

Figura 8.8. Valores C/R al ejecutar 6 millones de ficheros, usando 10 instancias y en un máximo de 3 horas, (a) en la Región EU (Frankfurt) y (b) en la Región EE.UU. (Norte de Virginia) de Amazon EC2 con Docker. 57

1. Introducción

Hoy en día, las empresas se enfocan en obtener datos de calidad, debido a que son recursos importantes y de gran utilidad a la hora de tomar decisiones. Para garantizar la calidad de los datos, estos deben ser analizados y transformados en datos consistentes, válidos, precisos y no redundantes.

Por ese motivo, se desarrollan aplicaciones que se encargan de extraer datos, reformatearlos, limpiarlos y visualizarlos mediante reportes, para que los clientes puedan tomar mejores decisiones y de esa manera aumentar la rentabilidad de su empresa. Una de estas aplicaciones es utilizada en este trabajo y tiene como objetivo procesar los datos de las ventas realizadas por las editoriales y librerías españolas.

Los datos que procesa esta aplicación, se encuentran en constante crecimiento y en algunos casos de forma exponencial, es decir, si hoy se procesa 1 GB de datos, el próximo mes se procesarán 2 GB y el siguiente 8 GB.

Para que la aplicación no tenga inconvenientes a la hora de procesar los volúmenes crecientes de datos, se requiere aumentar los recursos informáticos, lo que generaría un coste elevado para la empresa. También, podrían presentarse problemas de limitaciones en la infraestructura física, carencia de capital, falta de tiempo y de recursos informáticos para optimizar los sistemas existentes o simplemente los recursos implementados no son aprovechados al máximo, generando un gasto innecesario para la empresa.

La solución a estos problemas es el uso del Cloud Computing, que nos ofrece la posibilidad de adicionar o disminuir el número de máquinas virtuales que necesitemos para poder ejecutar la aplicación, lo que nos proporciona una gran escalabilidad. Además, sólo debemos realizar el pago por el tiempo que se encuentren activas las máquinas virtuales, con lo cual no realizaremos gastos innecesarios.

El Cloud Computing utiliza tecnologías de copias de seguridad, con lo cual la integridad de los datos que se procesan está garantizada. Otra característica importante es que los recursos informáticos siempre están disponibles, es decir, podemos acceder a ellos en cualquier momento.

Muchas empresas confían sus tareas empresariales de computación y almacenamiento a diferentes compañías que brindan el servicio de Cloud Computing [1], algunas de ellas son Amazon Web Services [2] , Rackspace [3] y Microsoft Azure [4].

Para el desarrollo de este trabajo se ha optado por utilizar los servicios de la empresa Amazon Web Services, porque es una de las primeras plataformas en la nube [5] y la más

popular en brindar servicios de Cloud Computing, además de contar con un servicio comercial muy estable.

Amazon EC2 es el servicio de AWS que nos ofrece un catálogo de diferentes tipos de instancias (máquinas virtuales), que podemos elegir de acuerdo a nuestras necesidades. Con la finalidad de mejorar la tolerancia a fallos, AWS utiliza centros de datos ubicados en diferentes regiones del mundo y de esa manera ofrece un servicio de calidad a sus clientes.

1.1. Objetivo de la investigación

Debido a la variedad de tipos de instancias que nos ofrece Amazon EC2, el objetivo de esta investigación es proporcionar y validar un modelo, que nos permita elegir el mejor tipo de instancia, para una ejecución óptima de la aplicación de gestión de editoriales y librerías en la infraestructura de la nube de Amazon Web Services.

Este modelo se realizará en base a los tiempos de ejecución total, al coste por uso de cada tipo de instancia y a la métrica de Coste/Rendimiento. El tipo de instancia que obtenga el menor valor de Coste/Rendimiento calculado, se considera la más óptima.

Para validar este modelo, la investigación se dividirá en dos casos, en el primer caso se utiliza un grupo de tipos de instancias de Amazon EC2, que han sido seleccionadas según sus características y en el segundo caso se utilizan los tipos de instancias utilizados en el primer caso y adicionalmente se incorpora el uso de la tecnología de contenedores Docker.

1.2. Estructura del trabajo

La memoria está organizada en nueve capítulos y se estructura de la siguiente forma. En el Capítulo 2 se detallan los conceptos de Cloud Computing, los tipos de servicios que ofrece, los modelos de despliegue, sus beneficios y riesgos.

El Capítulo 3 contiene información sobre Amazon Web Services, el servicio Amazon EC2 y los tipos de instancia que ofrece este servicio.

En el Capítulo 4 se detallan los conceptos de la tecnología Docker, su arquitectura, funcionamiento y las diferencias que existen con las máquinas virtuales.

En el Capítulo 5 se explica de forma detallada la aplicación ETL, su despliegue en las instancias Amazon EC2 y en Amazon EC2 con Docker.

En el Capítulo 6, se detalla brevemente la fórmula general del modelo, en base a los tiempos de ejecución y costos de cada una de los tipos de instancias EC2.

En el Capítulo 7, aplicando la fórmula general del modelo, se obtienen los resultados analíticos de los tiempos de ejecución y los valores de Coste/Rendimiento en Amazon EC2 y en Amazon EC2 con Docker.

El Capítulo 8 contiene algunos Casos de Uso, que nos mostrarán el desempeño de las instancias de Amazon EC2.

El Capítulo 9 muestra las conclusiones y el trabajo que sería interesante investigar en el futuro.

1. Introduction

Today, companies focus on obtaining quality data, because they are important and useful for taking decisions. To ensure the quality of data, these must be analyzed and transformed into consistent, valid, accurate and non-redundant data.

For this reason, there are developed applications for extracting data, reformat, clean and display them through reports, so customers can make better decisions and thus increase the profitability of your company. One of those applications is used in this work and aims to process data from sales made by Spanish publishers and bookstores.

Data processing this application are constantly growing and in some cases exponentially, for example, if today is processed 1 GB of data, next month will be processed 2 GB and the next 8 GB.

For the application does not have problems when processing growing volumes of data requires computer resources increase, which would generate a high cost to the company. There are also problems of limitations in physical infrastructure, lack of capital, lack of time and computer resources to optimize existing resources or simply the implemented resources are not maximized, generating an unnecessary expense for the enterprise.

The solution to these problems is the use of Cloud Computing, which gives us the ability to add or reduce the number of virtual machines that need to run the application, which provides high scalability. In addition, we only make payment for the time that virtual machines are active, which will not make unnecessary expenses.

Cloud Computing uses backup technologies, which the integrity of the data being processed is guaranteed. Another important feature is that computing resources are always available, so we can access them at any time.

Many companies entrust their enterprise computing and storage tasks to different companies providing Cloud Computing service [1], some of them are Amazon Web Services [2], Rackspace [3] and Microsoft Azure [4].

For the development of this work has chosen to use enterprise services Amazon Web Services, because it is one of the first cloud platforms [5] and the most popular in providing cloud computing services, in addition it has a service very stable business.

EC2 is an Amazon AWS service that offers a catalog of different types of instances (virtual machines), we can choose according to our needs. In order to improve fault tolerance, AWS uses data centers located in different regions of the world and thus offers a quality service to its customers.

1.1. Objective research

Due to the variety of instances offered by Amazon EC2, the objective of this research is to provide and validate a model that allows us to choose the best type of instance, for an optimal performance of a management application publishers and bookstores in the cloud infrastructure of Amazon Web Services.

This model will be based on total execution time, the cost per use of each type of instance and the metric of Cost/Performance. The type of instance that received the smallest value Cost/Performance calculated, is considered the most optimal.

To validate this model, research will be divided into two cases, in the first case a group of types of Amazon EC2 instances that have been selected according to their characteristics and in the second case the same instances for the first case are used and additionally the use of containers Docker technology is incorporated.

1.2. Document Structure

The memory is organized into nine chapters and is structured as follows. In Chapter 2, the concepts of Cloud Computing detailing the types of services offered, deployment models, its benefits and risks.

Chapter 3 contains information on Amazon Web Services, Amazon EC2 instance types and offering this service.

In Chapter 4, the concepts of Docker technology, architecture, performance and the differences with virtual machines are mentioned.

In Chapter 5, explained in detail the ETL application, its deployment in Amazon EC2 instances and Amazon EC2 with Docker.

In Chapter 6, briefly details the general formula model, based on the execution times and costs of each EC2 instance types.

In Chapter 7, applying the general formula model, the analytical results of the obtained execution times and Cost / Performance values on Amazon EC2 and Amazon EC2 with Docker.

Chapter 8 contains some use cases, which will show the performance of Amazon EC2 instances.

Chapter 9 shows the conclusions and the work that would be interesting to investigate in the future.

2. Cloud Computing

Se puede definir como un modelo que permite un cómodo acceso bajo demanda a un conjunto compartido de recursos informáticos configurables, por ejemplo: redes, servidores, almacenamiento, aplicaciones y servicios, que pueden ser rápidamente provisionados y liberados con el mínimo esfuerzo de administración o interacción con el proveedor de servicios [6].

El Cloud Computing es una alternativa muy rentable para poder desplegar nuestras aplicaciones, porque sólo se realizará el pago por el tiempo de uso de los recursos tecnológicos adquiridos.

Esta tecnología nos proporciona escalabilidad masiva, es decir, se tiene la posibilidad de aumentar o disminuir el número de instancias según la demanda que se tenga, es fiable, posee un alto rendimiento y es muy fácil de realizar la configuración de los recursos de computación, todo esto a un costo relativamente bajo en comparación con las infraestructuras dedicadas [7].

El único impedimento para tener una gran escalabilidad, se podría dar en el momento que se tengan limitaciones determinadas por razones financieras, es decir, no contar con los suficientes recursos económicos para poder adquirir un mayor número de máquinas virtuales.

Si no se tiene este tipo de limitaciones, entonces la escalabilidad se realizaría satisfactoriamente, a diferencia de los límites físicos que puedan aparecer al intentar adicionar nodos de clústeres o incluso supercomputadores en nuestras instalaciones físicas [8].

2.1. Tipos de servicios

Cloud Computing puede brindar tres tipos de servicios: IaaS, SaaS y PaaS. Infraestructura como Servicio (IaaS), este servicio le brinda al cliente almacenamiento, recursos de redes, disposición de procesamiento, con la finalidad de que el cliente sea capaz de desplegar y ejecutar sus propias aplicaciones en la infraestructura que le ofrece el proveedor.

En este tipo de servicio se puede escalar dinámicamente hacia arriba o hacia abajo, dependiendo de la necesidad del cliente y una de las ventajas más importante es que el

cliente solo tiene que pagar por el tiempo que utiliza este servicio [9] [10]. Por ejemplo: Amazon EC2, Rackspace, entre otros.

Plataforma como Servicio (PaaS), en este tipo de servicio el cliente tiene la capacidad de desplegar sus aplicaciones en una plataforma que el proveedor Cloud le brinda, utilizando los lenguajes de programación y herramientas soportadas por el proveedor [9].

Además, el cliente no tendrá que preocuparse por los servicios integrados que incluyen escalabilidad, mantenimiento, control de versiones y almacenamiento [10]. Por ejemplo: Force.com, hosting de páginas web y de correo electrónico, Microsoft Azure.

Software como Servicio (SaaS), este servicio le proporciona al cliente la capacidad de utilizar las aplicaciones del proveedor, que son ejecutadas en una infraestructura en la nube. Estas aplicaciones pueden ser accesibles desde los diferentes dispositivos del cliente, ya sea a través de una interfaz, por ejemplo un navegador web.

El usuario tiene la ventaja de no estar pendiente de la instalación, mantenimiento y almacenamiento, pero puede correr riesgos de seguridad y privacidad de sus datos [9].

Otra ventaja es que cuesta menos utilizar este tipo de servicio que realizar la compra de la aplicación. Es decir, el proveedor ofrece aplicaciones a un menor costo, las cuales son mucho más fiables para el cliente [10]. Por ejemplo: Google Apps, Salesforce.com, etc.

2.2. Modelos de despliegue

Pueden ser públicos, porque permiten el acceso de los usuarios a la nube a través de interfaces y los más usados son los navegadores web. En un Cloud público, los usuarios tienen que pagar sólo por el tiempo que utilizan el servicio, es decir, es un modelo de pago por uso.

Esto puede ser comparado con los servicios de agua o gas que recibimos en nuestras casas, pagamos sólo por la cantidad que utilizamos, ese mismo concepto se aplica en un Cloud público.

El Cloud público es menos seguro en comparación con los otros tipos de Cloud, debido a que todas las aplicaciones y datos se encuentran alojados en la nube pública, por ese motivo son más propensos a ataques maliciosos, pero para ello existe la solución de que los controles de seguridad se apliquen por el lado del proveedor Cloud y por el lado del cliente [11].

También puede ser privado, es decir, que el centro de datos sea interno, todo dentro de la misma organización. La principal ventaja es que es más fácil de administrar la seguridad,

el mantenimiento y las actualizaciones. También proporciona más control sobre el despliegue y uso de las aplicaciones, debido a que son gestionadas por la propia organización.

En comparación con la nube pública, donde todos los recursos y aplicaciones son gestionados por el proveedor de servicios, en la nube privada estos servicios se agrupan y están disponibles para los usuarios a nivel organizativo [11].

A la combinación de la nube pública y privada, se denomina nube híbrida. Este modelo es el más seguro para el control de los datos y aplicaciones. Además se le permite a la organización poder atender a sus necesidades en la nube privada y si se produce alguna necesidad ocasional, se le puede requerir a la nube pública recursos de computación intensiva.

2.3. Beneficios y riesgos

Para obtener los mejores resultados en el momento de hacer uso del Cloud Computing, debemos tener en cuenta los beneficios que nos ofrece esta tecnología y los riesgos que pueden presentarse [9].

Uno de los beneficios más importantes es que solo se realiza el pago por el tiempo que se utilizan los servicios.

Los proveedores de Cloud Computing, ofrecen los mejores niveles de servicio, infraestructuras mucho más fiables y una excelente disponibilidad de sus recursos en todo momento. Por ese motivo, los clientes tienen acceso de forma inmediata a los servicios.

Los clientes no tienen que preocuparse por comprar, instalar, ni probar el software o hardware que van a utilizar. Con ello, el personal de TI de los clientes, puede centrarse en el apoyo a la misión de la empresa y ya no estar pendiente de monitorear los recursos tecnológicos a utilizar.

El uso de los diferentes servicios del Cloud Computing puede variar, dependiendo de si sube o baja la demanda. Esto no resulta ser un problema, porque la escalabilidad se realiza simplemente apagando los recursos que no se desean utilizar o encendiendo los que se desean agregar.

Como riesgo podemos mencionar que los grandes proveedores, son objetivos para recibir ataques informáticos. Uno de ellos puede ser el ataque de autenticación, en donde el hacker tiene como objetivo acceder al sistema de la víctima.

Por lo general este acceso se realiza utilizando las sesiones establecidas por la víctima u obteniendo las credenciales de acceso. Esto es un motivo de preocupación para los clientes, porque ponen sus datos de propiedad en manos de los proveedores externos.

Utilizar los servicios de proveedores inmaduros o con nuevos modelos de negocio, puede ser un riesgo. Por ejemplo, los proveedores podrían optar por aumentar los precios, una vez que los clientes hayan contratado sus servicios.

Algunos problemas de rendimiento pueden ser originados por la latencia de la red o por inconvenientes en los centros de datos de los proveedores.

3. Amazon Web Services

Es un proveedor de Cloud Computing de la empresa Amazon, que ofrece un excelente servicio de computación en la nube, proporcionándoles a los clientes confidencialidad, integridad y disponibilidad de sus datos.

El servicio que brinda AWS es bajo demanda, es decir el cliente solo tiene que realizar el pago por el tiempo de uso de los recursos de computación, los cuales se encuentran disponibles a precios económicos [12] y no es necesario realizar ninguna inversión inicial para el uso de los mismos. El modelo de pago de AWS, es fijar el precio de la computación basado en horas de uso de las máquinas virtuales [13] [14].

Como podemos ver en la Figura 3.1, existe una gran variedad de productos ofrecidos por AWS y uno de los más utilizados es Amazon Elastic Compute Cloud (Amazon EC2), que brinda el servicio de crear diferentes tipos de instancias (máquinas virtuales), según la necesidad del cliente.

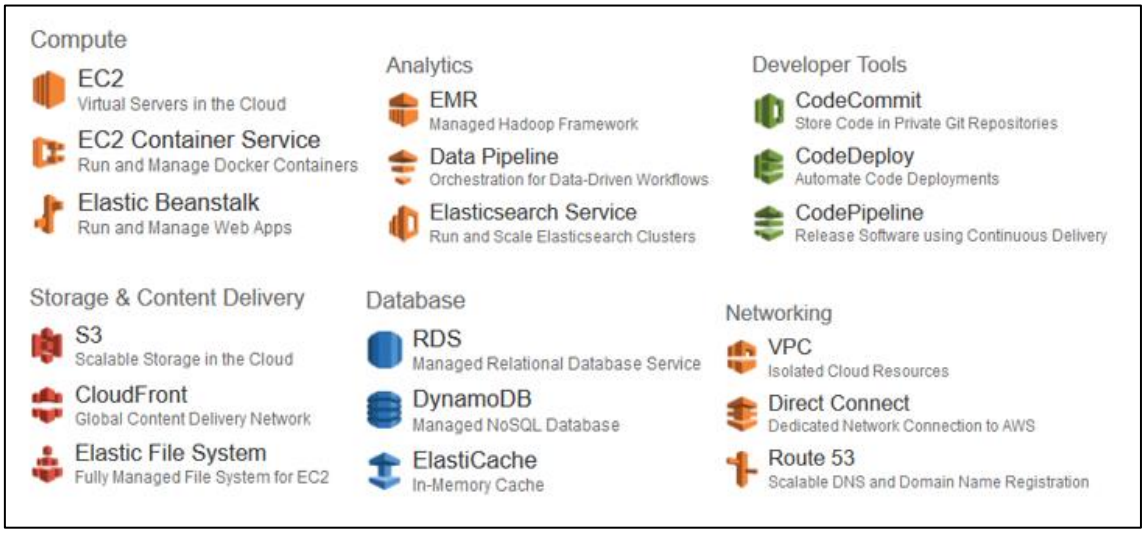

Figura 3.1. *Servicios que ofrece Amazon Web Services.*

Los diferentes tipos de instancias, son recursos virtuales que poseen características particulares de computación, de almacenamiento, de red y de la ejecución de un determinado sistema operativo, además los clientes pueden acceder a estos recursos virtuales a través de interfaces web [1].

3.1. Amazon Elastic Compute Cloud (Amazon EC2)

Este producto ofrece un entorno flexible de computación IaaS, utilizado para desplegar y ejecutar máquinas virtuales personalizadas [15], con opción a instalar diferentes sistemas operativos, aumentar las unidades de almacenamiento, configurar los tipos de conexiones, por ejemplo TCP, SSH, HTTP, entre otros. Con la libertad de realizar todo lo que se desea, debido a que se tiene asignado el acceso root.

La mayoría de los proveedores de servicios Cloud, utilizan técnicas de virtualización de máquinas para proporcionar recursos flexibles y rentables. Uno de ellos es Amazon EC2, ya que utiliza la virtualización Xen [16], que es un monitor de máquinas virtuales capaz de crear múltiples instancias en un único servidor físico, compartiendo procesadores físicos e interfaces de entrada y salida con otras máquinas virtuales, que se ofrecen a los clientes [17].

Amazon EC2 puede ser descrito como el más exitoso proveedor de computación en la nube IaaS y el servicio que ofrece es elástico, porque permite al usuario ampliar o reducir el tamaño de su infraestructura, mediante el incremento o disminución de las instancias [18][19].

Esta es una gran ventaja y no hay ningún problema en aumentar la escalabilidad según la demanda. Para los casos en los que se desee incrementar el número de instancias, utilizando la interface de Amazon EC2, este proceso es rápido y eficiente. De lo contrario, si la demanda baja, se pueden eliminar las instancias que no se desean utilizar, simplemente se apagan y en consecuencia dejaremos de pagar por ellas.

Por ese motivo, se puede decir que Amazon EC2 ha cambiado el modelo económico de la Informática, ya que solo se tendrá que pagar por el tiempo utilizado de las instancias desplegadas [20].

El pago se realiza por el número de horas que se ha utilizado la instancia. Si el tiempo de uso de una instancia es una hora y un segundo, entonces el pago se realizará por dos horas [21]. Además del tiempo de uso, el precio por intervalo de tiempo, varía en función de la configuración solicitada de la instancia (CPU, cantidad de memoria RAM, almacenamiento de instancia) [22].

Uno de los puntos más importantes a tener en cuenta al usar infraestructuras Cloud es la seguridad, por ese motivo dentro de Amazon EC2 la seguridad se proporciona en múltiples niveles: la configuración segura del sistema operativo que se instala en la instancia y de los servicios que se ejecutan en ella, el uso de servidores de seguridad y el uso de conexiones seguras utilizando el protocolo HTTPS para las comunicaciones entre clientes y las instancias [15].

En la práctica, las aplicaciones orientadas a servicios suelen utilizar diferentes centros de datos, para mejorar la tolerancia a fallos y ofrecer una buena calidad de servicio a los usuarios [23]. Por ese motivo, la infraestructura de la nube de AWS tiene 33 zonas de disponibilidad, distribuidas en 12 regiones geográficas en todo el mundo (Ver Figura 3.2).

Una región es una ubicación física en el mundo y están conformadas por varias zonas de disponibilidad, las cuales constan de uno o varios centros de datos, estas zonas ofrecen la capacidad de operar bases de datos y aplicaciones de producción con una disponibilidad, tolerancia a fallos y escalabilidad mayor que la que ofrecería un único centro de datos [24]. Para el desarrollo de este trabajo, se han utilizado las instancias de las regiones de Europa (Frankfurt) y de Estados Unidos (Norte de Virginia).

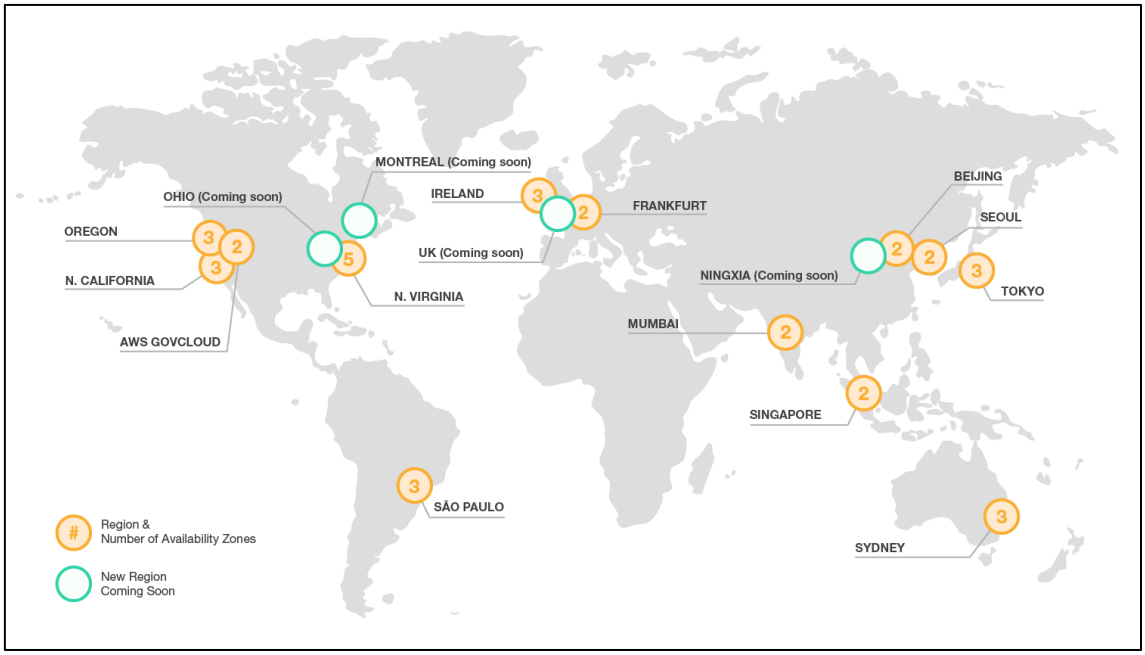

Figura 3.2. *Infraestructura global de Amazon Web Services.*
(Fuente: http://aws.amazon.com/es/about-aws/global-infrastructure).

Amazon EC2 nos ofrece la flexibilidad al elegir el hosting, ya que se puede elegir entre varios tipos de instancia y con diferentes sistemas operativos instalados. Amazon EC2 permite seleccionar un tamaño específico de memoria RAM, un tipo y tamaño de almacenamiento, además del tamaño de la partición de arranque del sistema operativo, que se puede elegir entre varias distribuciones de Linux y Microsoft Windows Server.

El control sobre las instancias es total, ya que se pueden detener las instancias sin perder los datos de sus respectivas unidades de almacenamiento y también pueden reiniciarse las instancias sin ningún inconveniente. Además, se tiene acceso a la información que brinda la consola de las instancias y todo ello a través del servicio web.

Es muy seguro, debido a que Amazon EC2 funciona junto con el servicio de Amazon VPC, con ello las instancias se ubican en una Virtual Private Cloud (VPC), para proporcionar una funcionalidad de red sólida y segura para sus recursos informáticos. El cliente decide las instancias que se exponen públicamente en Internet y las que se mantienen privadas [20].

3.1.1. Tipos de Instancia EC2

En Amazon EC2, las máquinas virtuales son llamadas instancias y este proveedor nos ofrece un catálogo de diferentes tipos de instancias, diferenciadas por las características de CPU, de la memoria RAM y por la E/S de disco. Estas instancias pueden ser utilizadas para el desarrollo de diferentes casos de uso, brindándole así al cliente la facilidad de poder elegir la instancia que mejor se adapte a su necesidad.

Las instancias de uso general T2, son ideales para las aplicaciones que no usan la CPU por completo, pero de vez en cuando tienen que alcanzar ráfagas basadas en el tiempo que se encuentran encendidas. Las instancias M3 que proporcionan un equilibrio de recursos informáticos, de memoria y red, serían una buena opción para muchas aplicaciones. Estos tipos de instancias pueden ser utilizadas para los casos de uso de bases de datos pequeñas y medianas, para entornos de desarrollo, para las tareas de procesamiento de datos que requieren memoria adicional y otras aplicaciones empresariales [25].

Las instancias con optimización informática C3 cuentan con la tecnología de los procesadores Intel Xeon E5-2680 v2 y han sido diseñadas para ejecutar aplicaciones que requieren un desempeño informático intensivo y las instancias C4 son la última generación de instancias con optimización informática, que disponen de los procesadores de mejor desempeño y ofrecen la mejor relación precio/desempeño informático de EC2. Este tipo de instancias pueden ser utilizadas para los casos de usos de flotas front-end de alto desempeño, el procesamiento por lotes, los análisis distribuidos, las aplicaciones científicas de alto desempeño y de codificación de vídeo [25].

Las instancias optimizadas para memoria R3, ofrecen un costo más bajo por GiB de RAM y son ideales para bases de datos de alto rendimiento, cachés de memoria distribuidas, análisis en memoria, Microsoft SharePoint y otras aplicaciones empresariales [25].

Las instancias optimizadas para almacenamiento I2 de E/S elevada, poseen una alta capacidad de almacenamiento por lo que ofrecen un almacenamiento de la instancia muy rápido respaldado por SSD y las instancias de GPU que se enfocan en ofrecer unidades de procesamiento de gráficos junto con un alto desempeño de CPU. Son ideales para realizar streaming de aplicaciones en 3D, codificación de video y otras cargas de trabajo de informática GPU o de gráficos del lado del servidor [25].

4. Docker

Es un proyecto de código abierto, que nos permite empaquetar las aplicaciones con todas sus dependencias, es decir, con todo lo que necesita para poder ejecutarse, como el código, las herramientas y las librerías del sistema, etc. A estos empaquetados se les denomina contenedores.

Estos contenedores pueden ser ejecutados en otra máquina que tenga instalado Docker, sin tener problemas con el entorno de ejecución.

Debido a que está basado en estándares abiertos, Docker se puede ejecutar en la mayoría de las principales distribuciones de Linux y en algunas distribuciones de Microsoft. Es seguro debido a que las aplicaciones de los contenedores están aisladas entre sí con lo cual le brinda una capa adicional de protección para cada aplicación [26].

Su uso es ligero, porque todos los contenedores que se ejecutan en una sola máquina están compartiendo el mismo núcleo del sistema operativo y así genera un uso eficiente de la memoria RAM.

Docker es una herramienta portátil, debido a que las imágenes de los contenedores requieren menos espacio de almacenamiento y en consecuencia el despliegue de los contenedores es rápido [27].

Las características principales de los contenedores son las siguientes: El tiempo de ejecución es rápido, la capacidad para desarrollar, probar e implementar las aplicaciones es muy eficiente y pueden interconectarse entre ellos [28].

4.1. Contenedor y Máquina virtual

Desde antes que aparezca el Cloud Computing, ya se utilizaba la virtualización para el aprovisionamiento de recursos de computación [29]. El enfoque arquitectónico de Docker es diferente al de las máquinas virtuales, siendo la portabilidad de Docker el punto más importante de su uso, debido a que son más fáciles de transportar. Como podemos ver en la Figura 4.1.a, una máquina virtual aloja todo el sistema operativo, las librerías, los bins necesarios y las diferentes aplicaciones, con lo cual se tendría que disponer de un gran tamaño de almacenamiento en la máquina anfitriona [26].

A diferencia de las máquinas virtuales, los contenedores Docker pueden ser vistos como herramientas más flexibles para el empaquetamiento, la entrega y el despliegue del software y de las aplicaciones.

Como podemos ver en la Figura 4.1.b, un contenedor está conformado por las librerías, los binarios necesarios y la aplicación con su respectiva dependencia, pero a diferencia de la máquina virtual, todos los contenedores comparten el mismo Kernel de la máquina que los aloja.

Cada uno de los contenedores posee su propio espacio de usuario en el sistema operativo anfitrión y se ejecutan en cualquier ordenador, en cualquier infraestructura y en cualquier nube que tenga desplegado el Docker Engine [26].

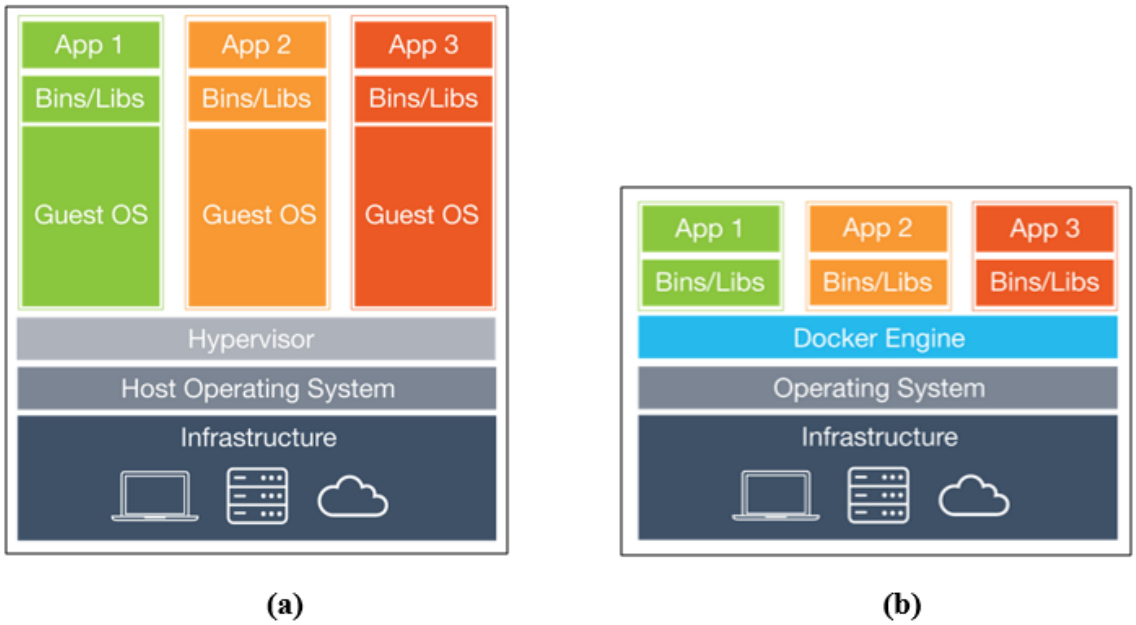

Figura 4.1. *Enfoque arquitectónico de (a) máquina virtual y (b) contenedor Docker. (Fuente: https://www.docker.com/what-docker)*

Debido a que los contenedores comparten el núcleo de Linux, son mucho más ligeros que las máquinas virtuales. En un ordenador típico, podrían funcionar pocas máquinas virtuales a la vez, pero no se tendrían problemas para ejecutar 100 contenedores Docker. Esta característica ha hecho que Docker sea atractivo para el sector empresarial y aumente su inmensa popularidad [30].

El núcleo de los contenedores se basa en los namespaces y cgroups de Linux, con ello se permite aislar el conjunto de los puntos de montaje del sistema de ficheros [31]. Es decir, dentro de un contenedor, se puede ejecutar una aplicación cuyos namespaces, son diferentes a los namespaces de otro contenedor que se despliegue en la misma máquina anfitriona.

4.2. Arquitectura de Docker

La arquitectura es la de cliente-servidor, el cliente se comunica con el demonio de Docker, que sería la parte servidor y se encarga de la construcción, funcionamiento y distribución de los contenedores. En la gran mayoría de los casos, el cliente y el demonio de Docker, se pueden ejecutar en el mismo sistema o también conectar un cliente a un demonio Docker que se encuentre a distancia, esta comunicación se realiza a través de sockets o de una API REST.

Como podemos ver en la Figura 4.2, el demonio de Docker actúa como el servidor central, que trabaja con las imágenes y los contenedores. El cliente se comunica con el demonio, enviándole órdenes a través de comandos [32].

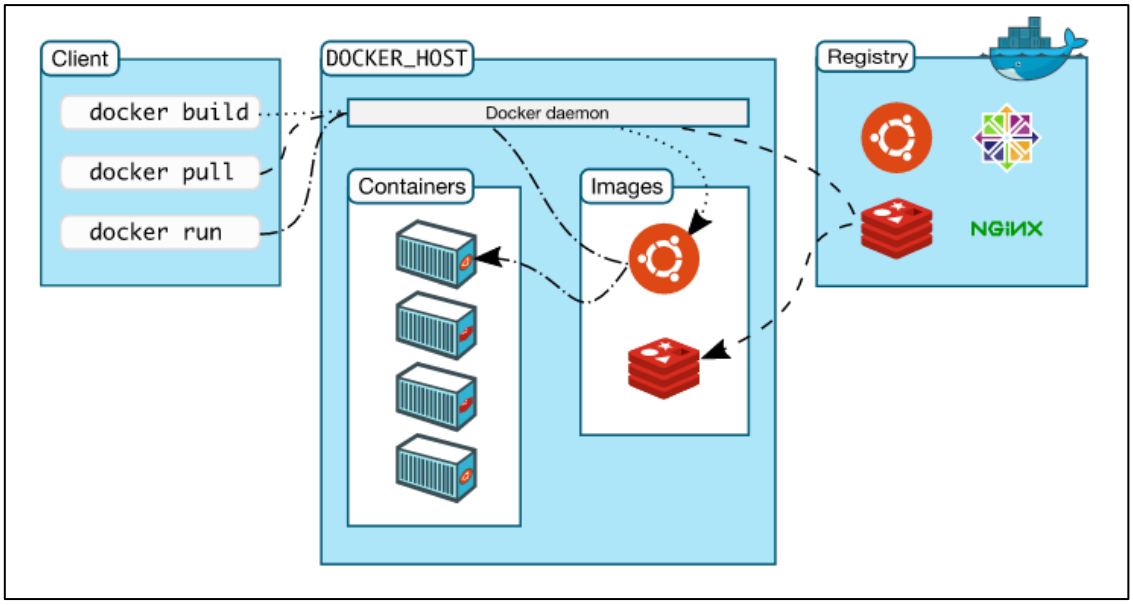

Figura 4.2. *Arquitectura de Docker.*

En la arquitectura de Docker, existen tres recursos muy importantes, que son las imágenes, los registros y los contenedores de Docker.

Las imágenes de Docker, son plantillas de sólo lectura y es el punto de partida para crear contenedores, es decir, desde una imagen de Docker, se pueden generar miles de contenedores y cada uno de ellos se encuentra completamente aislado.

Docker ofrece una forma sencilla de construir nuevas imágenes, actualizar las imágenes existentes o descargar las imágenes Docker que otros usuarios han creado.

Existen muchas comunidades que alojan en los repositorios del Docker Hub, las imágenes de sus respectivos softwares. Por ejemplo, podemos encontrar imágenes oficiales de Ubuntu, CentOS, Fedora, etc. con aplicaciones como Apache, Nginx, PHP, Java ya

instaladas o también se pueden encontrar imágenes que han sido creadas por otros usuarios.

Una diferencia clave entre las imágenes Docker y las máquinas virtuales, es que las imágenes Docker comparten el núcleo de Linux con la máquina anfitriona. Para el usuario final, la consecuencia principal de esto, es que cualquier imagen Docker, debe basarse en un sistema compatible con Linux.

Docker ofrece el servicio Docker Hub [33], que es el componente oficial de la distribución de imágenes de Docker registradas y la creación de una cuenta es gratuita.

En Docker Hub, se alojan diferentes imágenes, las cuales pueden ser públicas o privadas. Si es pública, cualquier usuario puede acceder a las imágenes que hemos creado, caso contrario sería tener la cuenta en modo privado, pero para utilizar esta opción se necesita pagar. Existen una diversidad de imágenes, las que uno mismo crea, las que otros crean y las de las diferentes comunidades, estas últimas son consideradas como repositorios oficiales.

A partir de una imagen de Docker, se pueden crear un gran número de contenedores, los cuales se encuentran completamente aislados y seguros. En los contenedores Docker, se instala y configura todo lo que se necesita para que las aplicaciones se ejecuten satisfactoriamente.

Se debe generar una imagen a partir del contenedor y posteriormente subirlo al Docker Hub, para que otros usuarios puedan descargar la imagen y generar los contenedores deseados [32]. Al igual que las máquinas virtuales, después de generarse los contenedores de Docker, éstos se pueden iniciar, detener, reiniciar, pausar y si se desea también pueden ser eliminados.

4.3. Contenedores como Servicio (CaaS)

Como se indicó en la sección 2.1, los tipos de servicios que ofrece el Cloud Computing son Infraestructura como Servicio (IaaS), Plataforma como Servicio (PaaS) y Software como Servicio (SaaS), pero Docker nos ofrece Contenedores como Servicio (CaaS), que nos proporciona agilidad, control y portabilidad, para construir un gran número de contenedores, en los cuales podemos alojar diferentes aplicaciones que puedan ser ejecutadas en cualquier lugar [34].

El uso del servicio CaaS de Docker, nos ofrece la libertad de definir el entorno en el que se van a utilizar los contenedores. De esa manera, poder crear y desplegar las aplicaciones de manera más rápida y fácil, con lo cual se podrá responder rápidamente a los cambios que se presenten.

Los usuarios tienen el control total, debido a que cuentan con la capacidad de administrar y operar de forma segura la escalabilidad de los contenedores. Además, Docker ofrece portabilidad en todo el ciclo de los contenedores, porque pueden ser desplegados en diferentes entornos que soporten esta tecnología.

4.4. Funcionamiento de Docker

En la Figura 4.3, se puede observar cómo se encuentra estructurado Docker, su funcionalidad, cómo se elaboran, cómo se comparten los contenedores y en qué momento se ejecutan.

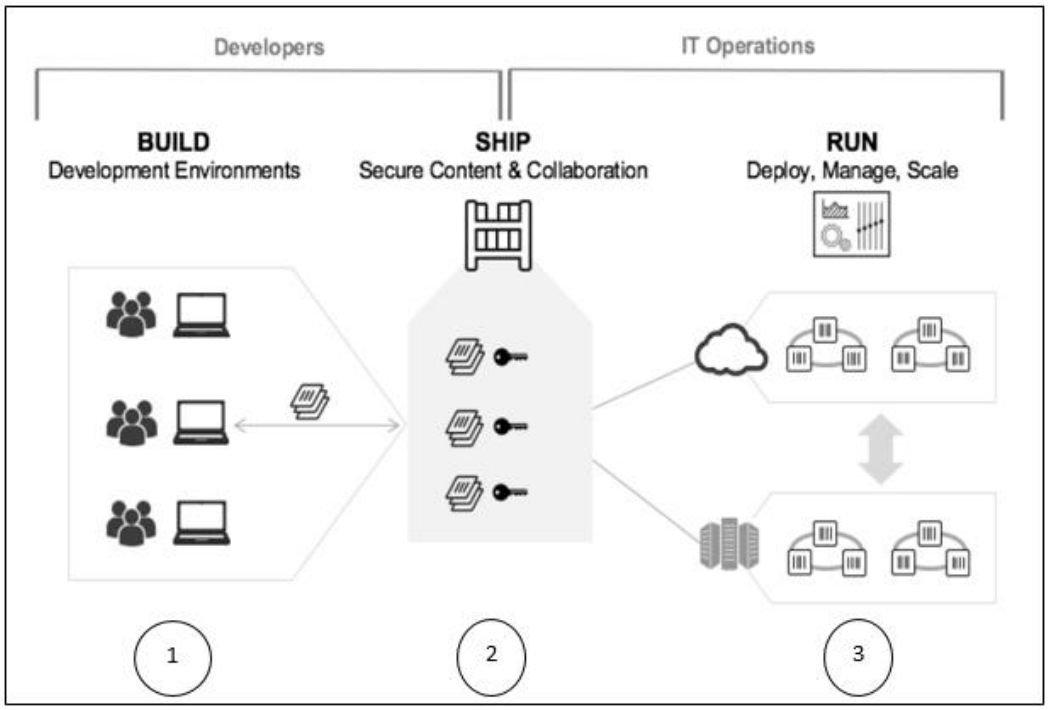

Figura 4.3. *Esquema del funcionamiento de Docker.*

El funcionamiento de Docker, está conformado por tres pasos importantes, que son construir, enviar y ejecutar. En el paso de construir, Docker nos permite crear y desplegar nuestras aplicaciones, en los entornos de desarrollo y producción, sin preocuparnos de la plataforma o el lenguaje en que se encuentren desarrolladas las aplicaciones.

Como base se utilizan las imágenes que se encuentran alojadas en los repositorios de Docker. Para la creación de una imagen de Docker, se puede utilizar Dockerfile, que es un fichero en el cual se describe todo el procedimiento, mediante una serie de instrucciones con las cuales se crea una imagen. La ejecución del Dockerfile, se inicia con la descarga de la imagen base desde el servidor Docker Hub, en caso de que no se

tenga descargada la imagen. Luego se procede a añadir los paquetes de software a instalar y si se desea la ejecución de comandos adicionales.

La construcción de una imagen utilizando un Dockerfile, es comparable a la instalación de una nueva máquina desde cero, que puede tomar desde minutos hasta horas, dependiendo del número de capas implicadas. Sin embargo, el tamaño de un Dockerfile es bastante pequeño (de 1 KB o menos) y se transfiere fácilmente a través del sistema [35]. También existe la opción de crear imágenes de Docker, a partir de contenedores desplegados.

El segundo paso es enviar o registrar nuestra imagen. Luego de haber creado nuestra imagen, podemos alojarla en Docker Hub (ver Sección 4.5), que es el servidor que Docker nos ofrece. Si es que nuestra cuenta es pública, nuestras imágenes están disponibles para todos los usuarios. Finalmente, para ejecutar un contenedor solo se necesita tener descargada la imagen que necesitamos como base.

Docker puede ejecutar cualquier aplicación y en cualquier lugar, ya que es independiente del entorno en que se desplieguen los contenedores [34] .

4.5. Herramientas de Docker

Docker nos ofrece muchas herramientas y las más importantes son las siguientes [36]. Docker Engine, que es la herramienta principal, porque ofrece las funciones básicas para la creación de las imágenes y contenedores de Docker.

La herramienta Docker Machine, se utiliza para instalar Docker Engine en uno o más sistemas virtuales, pueden ser locales o remotos. Esta herramienta es muy utilizada por los usuarios que tienen el sistema operativo Windows o OS X.

Docker Hub, es el servidor de Docker, en donde se alojan las imágenes creadas por diferentes usuarios y desde este servidor se descargan las imágenes base para crear los contenedores. Para implementar un clúster nativo de varios contenedores de Docker, se utiliza Docker Swarm.

Docker Compose, es una herramienta que define y ejecuta varios contenedores de Docker utilizando un archivo .yml, en el cual se definen todas las configuraciones de los contenedores que se van a ejecutar.

Docker nos ofrece varias herramientas y para el desarrollo de este trabajo de todas las que se han detallado, se utilizaron Docker Machine, Docker Engine y Docker Hub.

5. Aplicación ETL

Para el desarrollo de este trabajo, se ha utilizado una aplicación ETL (Extracción, Transformación y Carga), que ha sido creada por una empresa dedicada a la minería de datos y fue desarrollada utilizando la herramienta Kettle de Pentaho.

Un proceso ETL está conformado por tres pasos consecutivos, ya que un paso depende del anterior [37]. El primer paso se encarga de la extracción eficaz de los datos desde los sistemas de origen para el proceso ETL. La transformación, es el segundo paso en cualquier escenario ETL. Este paso se encarga de realizar la limpieza de datos, transformación e integración de los datos de entrada, con el objetivo de obtener datos precisos, correctos, consistentes y sin ambigüedades. El último paso es la carga de los datos extraídos y transformados a una base de datos final.

La aplicación ETL utilizada, tiene como objetivo extraer datos (en este caso de las ventas realizadas por las diferentes editoriales y librerías españolas), limpiar y transformar estos datos y por último cargarlos en una base de datos.

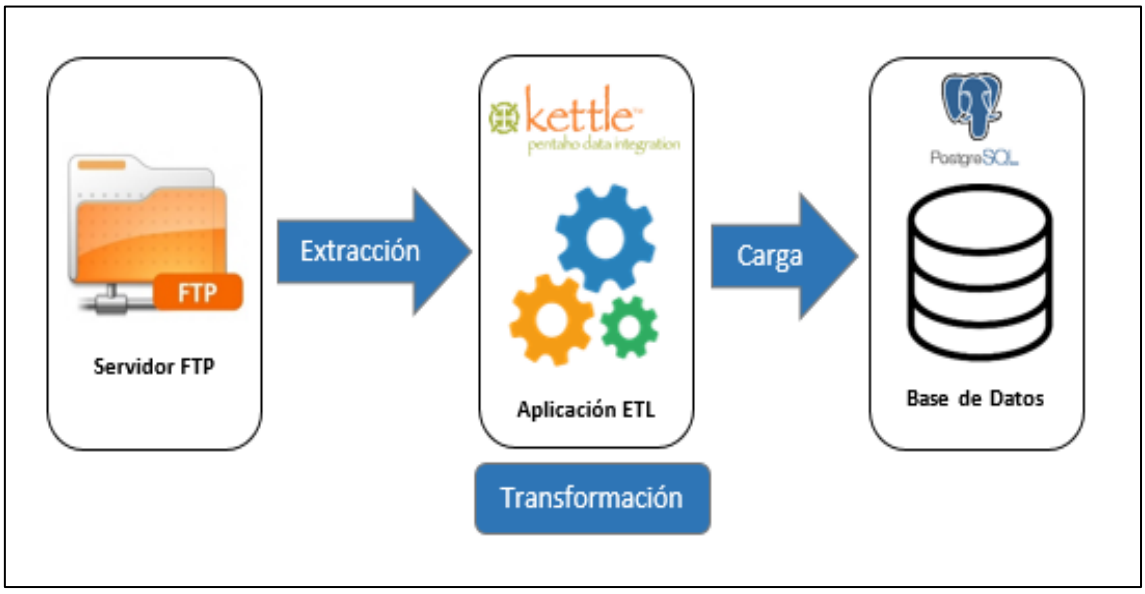

Figura 5.1. *Esquema de la Aplicación ETL.*

En la Figura 5.1, se puede observar el esquema del proceso de la Aplicación ETL que realiza la aplicación en el ambiente local. El primer paso es extraer un conjunto de ficheros que se encuentran alojados en un servidor FTP, todos estos ficheros son brindados diariamente por diferentes proveedores especializados en gestión de ventas de editoriales y librerías.

Luego se realiza el proceso de transformación, en el que se aplica un conjunto de reglas de negocio sobre los datos extraídos, con la finalidad de limpiar o modificar el formato de estos datos según sea necesario, y finalmente cargarlos en una base de datos PostgreSQL.

Con los datos cargados se elaboran reportes estadísticos que son enviados a los clientes, con la finalidad de que se puedan tomar mejores decisiones, con ayuda de la información brindada en los reportes.

En el ambiente local, la aplicación ETL se ejecuta en un equipo que tiene un procesador Intel(R) Xeon(R) CPU E5-2650 v2@ 2.60 GHz, con memoria RAM de 2GB y tiene instalado el sistema operativo Microsoft Windows Server 2003 Standard Edition Service Pack 2. En este equipo local, también se realiza la ejecución de otras aplicaciones que posee la empresa.

En este trabajo, el tiempo de ejecución que se tuvo en cuenta, fue el tiempo transcurrido desde el inicio al final de la ejecución del proceso ETL, es decir, el tiempo de extracción, transformación y carga de los datos.

5.1. Aplicación ETL en Amazon EC2

Amazon EC2 nos brinda la posibilidad de crear y utilizar instancias EC2 en cualquiera de las diferentes regiones, que están dispersadas geográficamente.

Para el desarrollo de este trabajo se han utilizado instancias de dos regiones: Región EE.UU. Este (Norte de Virginia) y Región de la UE (Frankfurt).

Se eligieron estas dos regiones, para poder determinar si existe una gran diferencia en los tiempos de ejecución de la aplicación ETL en diferentes continentes. Además ambas regiones tienen diferencias en el precio por hora, siendo la Región de EE.UU. Este (Norte de Virginia), la que posee los precios más baratos.

Teniendo en cuenta lo indicado en la sección 3.2.1, la aplicación ETL se desplegó en las instancias de uso general (t2.small, t2.medium, m3.medium y m3.large), en la instancia con optimización informática (c4.large) y en la instancia optimizada para memoria (r3.large).

En las Tablas 1 y 2, se detallan las características de las instancias utilizadas por región, donde ECU significa Unidad de Computación EC2 y cada ECU provee la capacidad de una CPU a 1.0-1.2Ghz de 2007 (Xeon/Opteron). Todas las instancias utilizadas, tenían instalado el sistema operativo Ubuntu Server 14.04 LTS Trusty Tahr.

Se puede observar que las características en ambas regiones son similares y sólo existe diferencia en el precio por hora.

Tabla 1. *Características de las instancias que ofrece Amazon EC2 en la Región EE.UU. Este (Norte de Virginia).*

Región EE.UU. Este (Norte de Virginia)					
Instancia EC2	vCPU	ECU	Memoria	Plataforma	Precio/hora
t2.medium	2	Variable	4GB	64 bit	$0.052
m3.medium	1	3	3,75GB	64 bit	$0.067
m3.large	2	6,5	7,5GB	64 bit	$0.133
c4.large	2	8	3,75GB	64 bit	$0.105
r3.large	2	6,5	15GB	64 bit	$0.166

Región UE (Frankfurt)					
Instancia EC2	vCPU	ECU	Memoria	Plataforma	Precio/hora
t2.medium	2	Variable	4GB	64 bit	$0.06
m3.medium	1	3	3,75GB	64 bit	$0.079
m3.large	2	6,5	7,5GB	64 bit	$0.158
c4.large	2	8	3,75GB	64 bit	$0.134
r3.large	2	6,5	15GB	64 bit	$0.2

Con los tiempos de ejecución de la aplicación ETL obtenidos en cada una de las instancias, se realizaron análisis comparativos, además de tener en cuenta el costo de cada instancia.

5.2. Aplicación ETL en Amazon EC2 con Docker

Para realizar las pruebas utilizando la tecnología Docker, se realizó la instalación de Docker Engine, en cada una de las instancias que se muestran en las Tablas 1 y 2.

La aplicación ETL se alojó en una imagen de Docker, para crear esta imagen se utilizó un archivo Dockerfile, donde se indicó que la imagen base es Ubuntu 14.04, además de las actualizaciones e instalación de las dependencias de la aplicación ETL. La imagen de Docker creada se alojó en el servidor Docker Hub y en cada una de las instancias se realizó el despliegue de un contenedor de Docker, que contiene la aplicación ETL.

Por último, se obtuvieron los tiempos de ejecución de la aplicación ETL de los contenedores de Docker desplegados y se calcularon los valores de Coste/Rendimiento, con estos datos se eligió la mejor instancia.

6. Modelo de Ejecución

Para obtener la fórmula del modelo de ejecución, se realizaron diferentes pruebas experimentales en cada una de las instancias seleccionadas de ambas regiones, tanto para el caso de ejecución en Amazon EC2, como para el caso de ejecución en Amazon EC2 con la tecnología Docker.

Con las líneas de regresión que se generan en base a los tiempos de ejecución obtenidos en las pruebas experimentales, con el precio por hora que podemos obtener de las Tablas 1 y 2, y con el número de instancias utilizadas, se realizó la formulación del modelo de ejecución completo.

6.1. Resultados experimentales

Para obtener los tiempos de ejecución de la aplicación ETL, se generaron diferentes intervalos de carga de datos, los cuales fueron establecidos en paquetes de 50, 100, 250, 500, 1.000, 1.750, 2.500, 5.000, 10.000, 12.500, 15.000, 20.000 y 25.000 ficheros.

6.1.1. En Amazon EC2

En las gráficas de las Figuras 6.1 y 6.2 se muestran las líneas de regresión, en base a los tiempos de ejecución que se han obtenido en cada una de las instancias de Amazon EC2 y de ambas regiones.

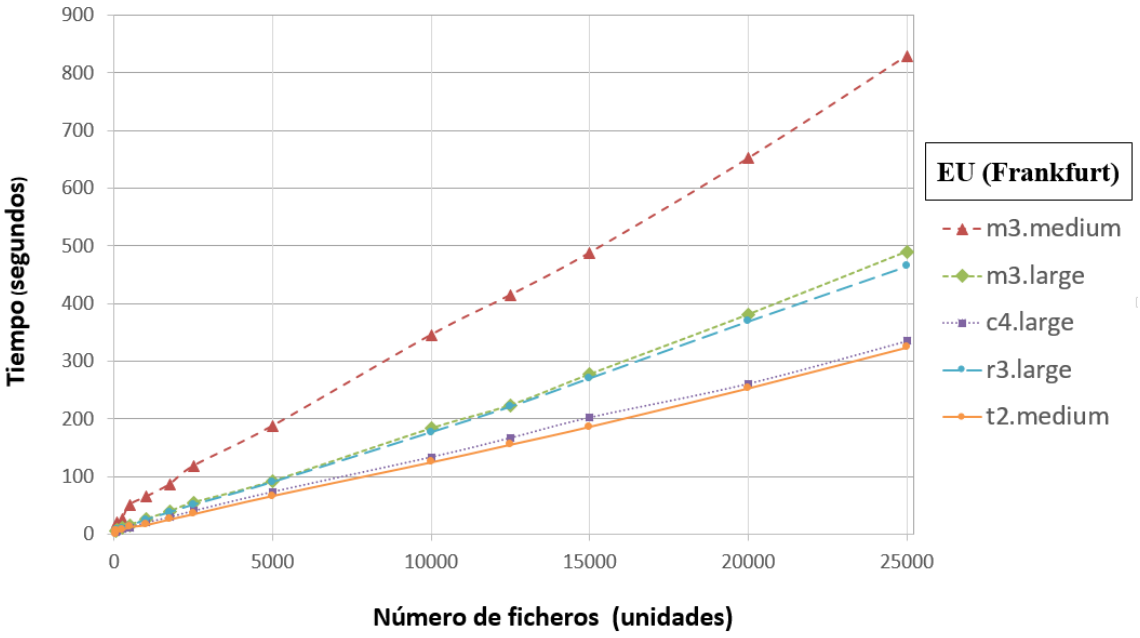

Figura 6.1. *Tiempos de ejecución en la región EU (Frankfurt) de Amazon EC2.*

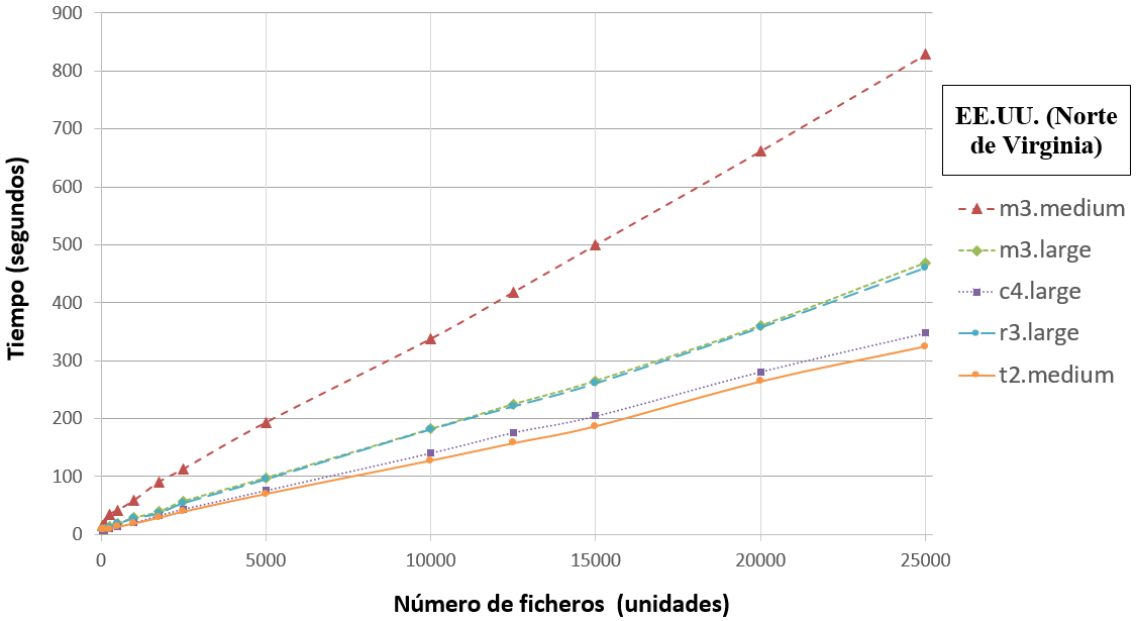

Figura 6.2. *Tiempos de ejecución en la región EEUU Este (Norte de Virginia) de Amazon EC2.*

Además teniendo en cuenta los datos de las Tablas 1 y 2, se puede deducir que en ambas regiones, la instancia m3.medium, tuvo el peor tiempo de ejecución por un amplio margen, esto no es sorprendente, teniendo en cuenta sus capacidades, que son relativamente bajas. La instancia m3.medium posee un core menos que los demás, tiene el menor número de unidades de computación ECU y una menor cantidad de memoria RAM.

Las instancias m3.large y r3.large casi obtienen los mismos tiempos de ejecución en ambas regiones, esto se debe a que sus características son similares. La instancia r3.large tiene el doble de memoria RAM que la instancia m3.large, por ese motivo la instancia r3.large obtiene mejores tiempos que la instancia m3.large, pero la diferencia es mínima.

Las instancias c4.large y t2.medium, obtienen los mejores tiempos de ejecución en ambas regiones. Además sus tiempos son casi iguales y esto se debe a que tienen las mejores características de cómputo.

A diferencia de la instancia c4.large, la instancia t2.medium es un tipo de instancia que su desempeño de rendimiento está basado en ráfagas, por ese motivo obtiene los mejores tiempos de ejecución de todas las instancias.

En ambas regiones, todas las líneas de regresión de una misma instancia, son similares y mantienen su posición en cuanto a su rendimiento. Si se desea obtener los tiempos estimados de cada una de las instancias EC2, para cualquier otro intervalo de número de ficheros, se debe utilizar la fórmula que se obtiene de las líneas de regresión.

Diferencia con la máquina local

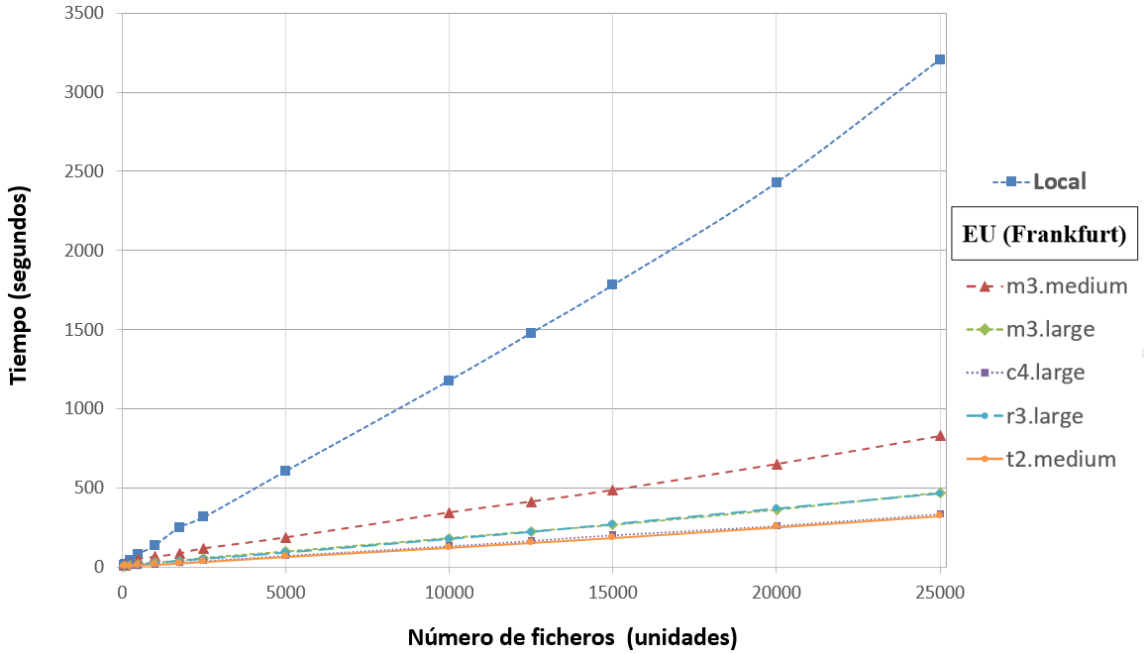

Figura 6.3. *Tiempos de ejecución en local y en la región EU (Frankfurt) de Amazon EC2.*

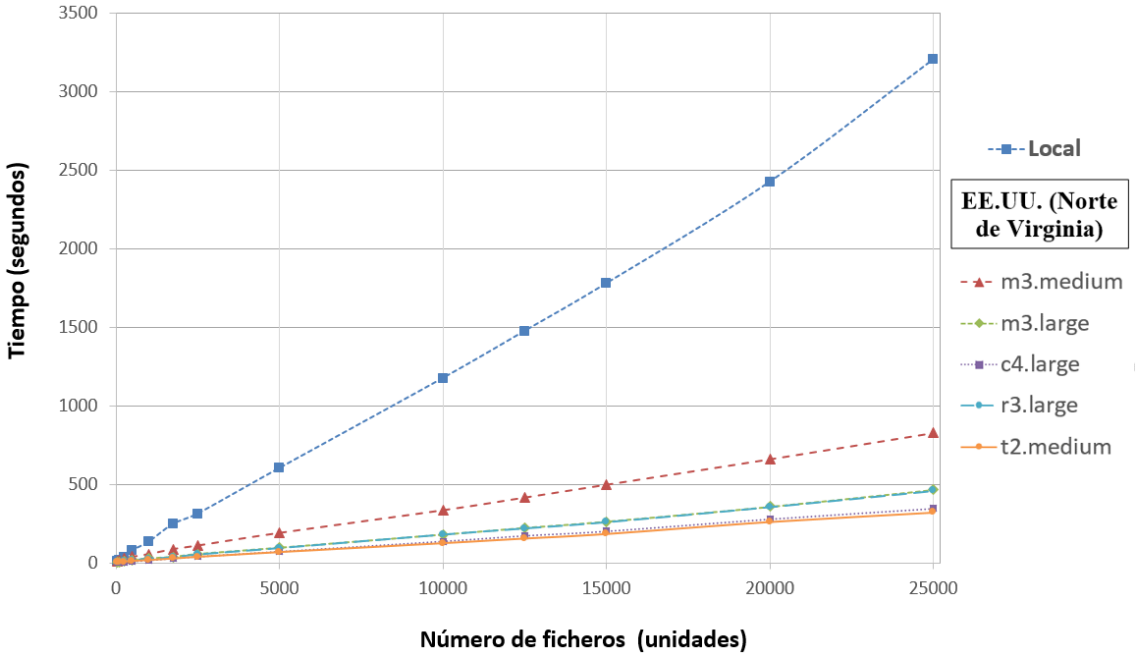

Figura 6.4. *Tiempos de ejecución en local y en la región EEUU Este (Norte de Virginia) de Amazon EC2.*

En las figuras 6.3 y 6.4, se pueden observar la gran diferencia en los tiempos de ejecución que existe entre la máquina local y las instancias EC2 de ambas regiones.

Mientras se incrementa el número de ficheros a ejecutar, es evidente que la diferencia entre los tiempos de ejecución de la máquina local y las instancias EC2 aumenta considerablemente.

Por ejemplo, si analizamos los datos de la gráfica de la Figura 6.4, para la ejecución de la aplicación ETL en el intervalo de 25.000 ficheros: el tiempo obtenido por la máquina local es de 3.206 segundos y el peor tiempo obtenido de todas las instancias EC2 es 829 segundos y le corresponde a la instancia m3.medium. Es decir, el tiempo obtenido por la máquina local es aproximadamente el cuádruple del tiempo obtenido por la instancia m3.medium, teniendo en cuenta que siendo aún la instancia que ofrece el peor tiempo, nos brinda un mejor tiempo de ejecución que la máquina local.

Observando las gráficas de ambas regiones y con el ejemplo anterior, se puede deducir que ejecutando la aplicación en la máquina local, no se obtendrán mejores tiempos que ejecutándola en las instancias EC2.

6.1.2. En Amazon EC2 con Docker

En las gráficas de las Figuras 6.5 y 6.6 se muestran las líneas de regresión, en base a los tiempos de ejecución que se han obtenido en cada una de las instancias de Amazon EC2 usando la tecnología Docker y de ambas regiones.

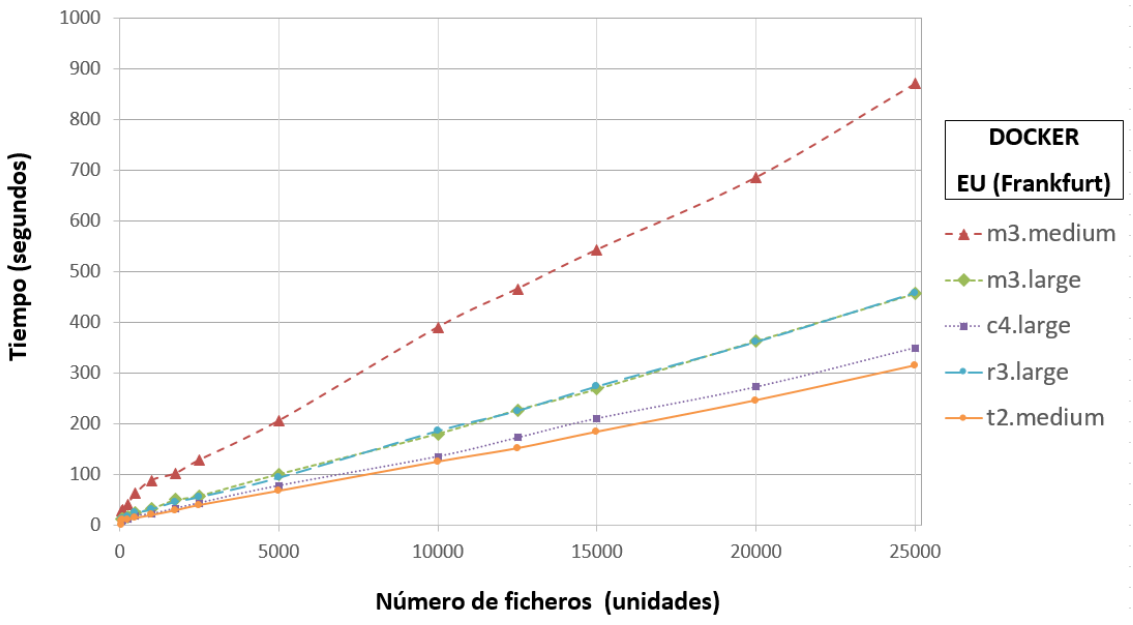

Figura 6.5. *Tiempos de ejecución en la región EU (Frankfurt) de Amazon EC2 con Docker.*

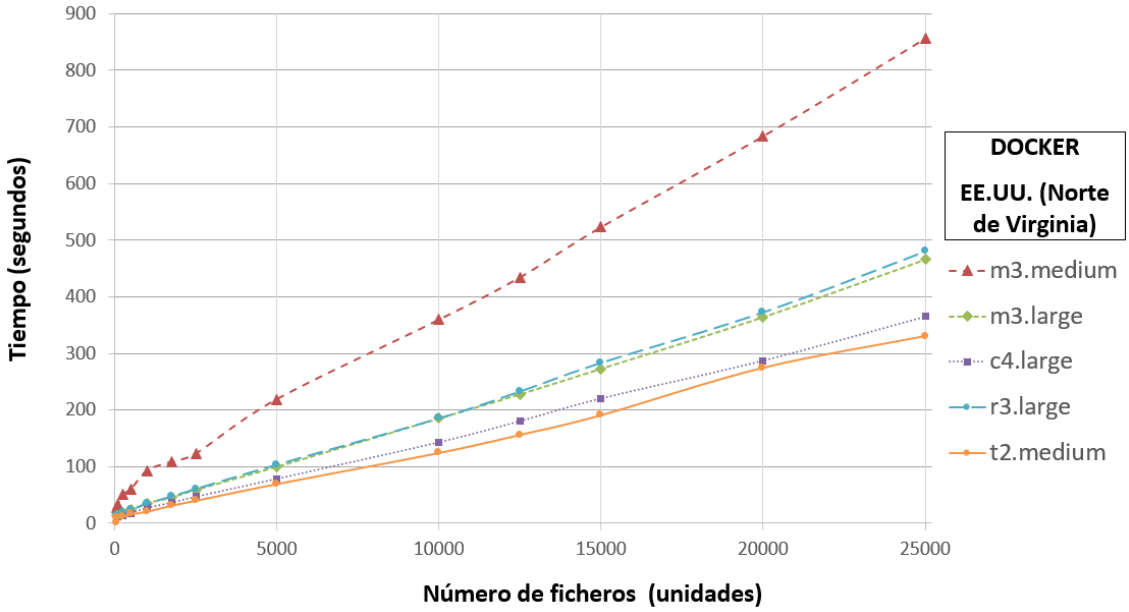

Figura 6.6. *Tiempos de ejecución en la región EEUU Este (Norte de Virginia) de Amazon EC2 con Docker.*

Además, teniendo en cuenta los datos de las Tablas 1 y 2, se puede deducir que al igual que los resultados obtenidos en la sección 6.1.1, la instancia m3.medium ofrece el peor tiempo de ejecución, debido a que sus capacidades de cómputo, son las más bajas, con respecto al resto de las instancias.

La instancia m3.medium, posee un core menos que los demás, tiene el menor número de unidades de computación ECU y una menor cantidad de memoria RAM.

Debido a que las instancias m3.large y r3.large, poseen características similares, los tiempos de ejecución obtenidos en ambas regiones son casi iguales, pero por el motivo que la instancia m3.large tiene la mitad de memoria RAM que la instancia r3.large, esta última obtiene mejores tiempos de ejecución que la instancia m3.large, aunque es mínima la diferencia.

Las instancias c4.large y t2.medium, obtienen los mejores tiempos de ejecución en ambas regiones. Además sus tiempos son casi iguales y esto se debe a que tienen las mejores características de cómputo.

La instancia t2.medium es un tipo de instancia que su desempeño de rendimiento está basado en ráfagas, por ese motivo obtiene los mejores tiempos de ejecución de todas las instancias.

En ambas regiones, todas las líneas de regresión de una misma instancia, son similares y mantienen su posición en cuanto a su rendimiento.

Con las líneas de regresión de las gráficas, se pueden obtener los tiempos de ejecución estimados de cada una de las instancias EC2 con Docker, para cualquier otro intervalo de número de ficheros.

Diferencia con la máquina local

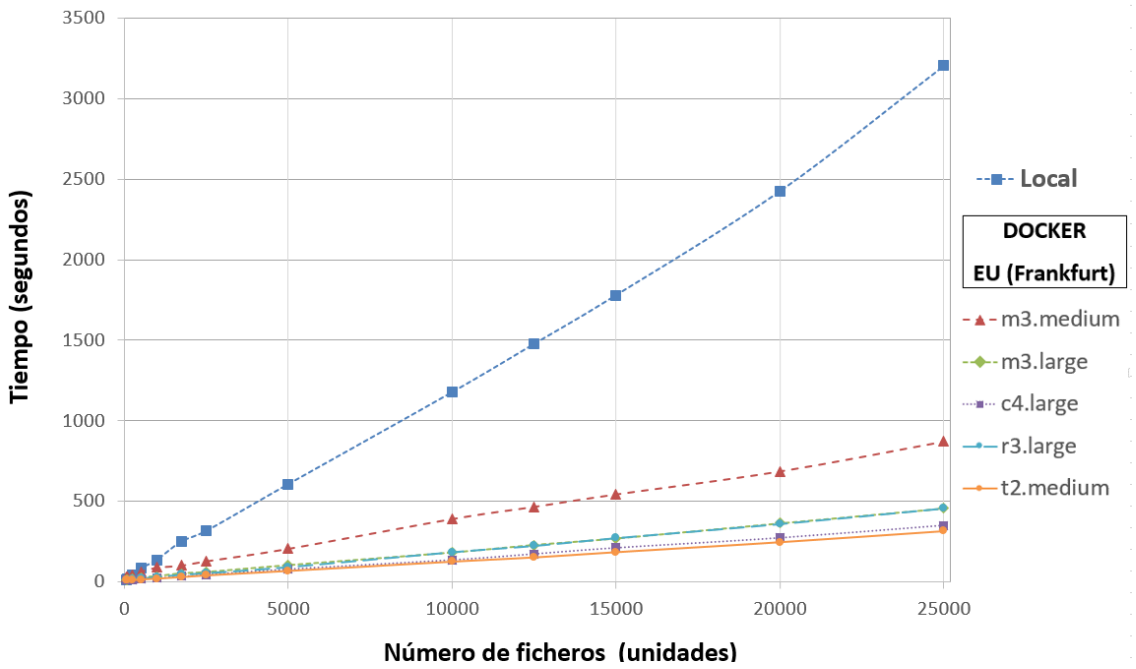

Figura 6.7. *Tiempos de ejecución en local y en la región EU (Frankfurt) de Amazon EC2 con Docker.*

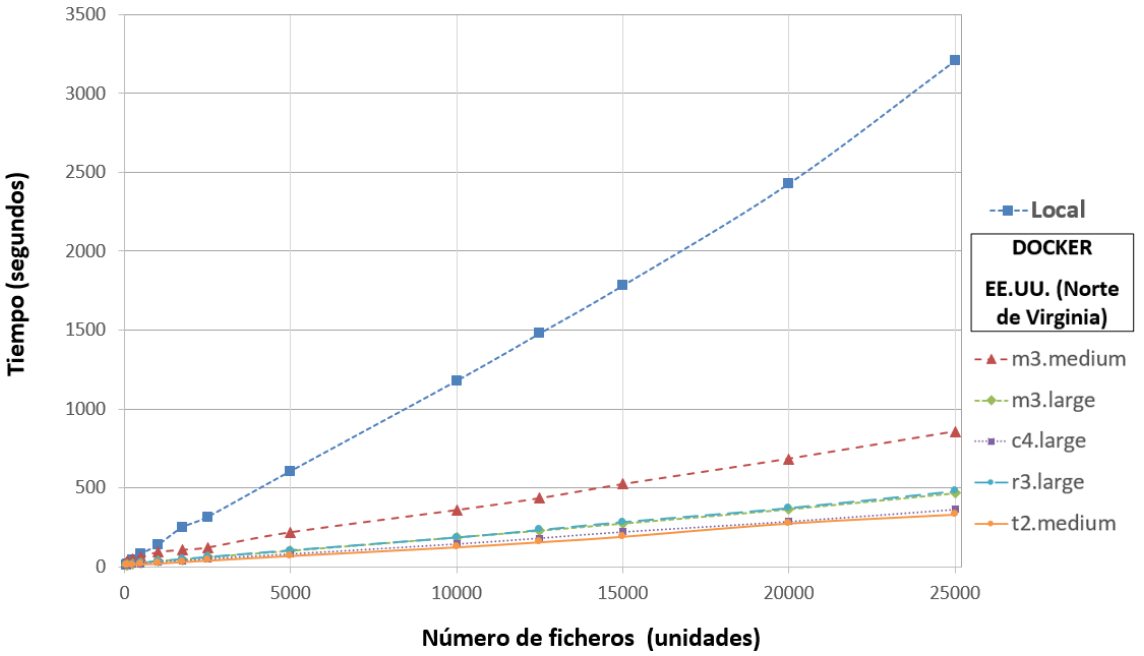

Figura 6.8. *Tiempos de ejecución en local y en la región EEUU Este (Norte de Virginia) de Amazon EC2 con Docker.*

En las figuras 6.7 y 6.8, se pueden observar que en ambas regiones, los tiempos de ejecución de las instancias EC2 usando Docker, son mucho mejores que los tiempos de ejecución que ofrece la máquina local.

Al igual que los resultados obtenidos en la sección 6.1.1, mientras el número de ficheros a ejecutar aumenta, podemos observar que la diferencia entre los tiempos de ejecución de la máquina local y de las instancias EC2 usando Docker, se incrementa considerablemente.

Por ejemplo, si analizamos los datos de la gráfica de la Figura 6.7 en la región EU (Frankfurt), para la ejecución de la aplicación ETL en el intervalo de 20.000 ficheros: la máquina local ha obtenido un tiempo de ejecución de 2427,08 segundos y el mejor tiempo de ejecución de todas las instancias es de 245,86 y le corresponde a la instancia t2.medium. Es decir, el tiempo obtenido por la instancia t2.medium es aproximadamente 10 veces mejor que el tiempo de ejecución de la máquina local.

Observando las gráficas de ambas regiones y con el ejemplo anterior, se puede deducir que ejecutando la aplicación ETL en las instancias EC2 y usando Docker, se han obtenido mejores tiempos que en la máquina local.

6.2. Fórmula del Modelo

Después de haber analizado los tiempos de ejecución de cada una de las instancias, se desea saber cuál de ellas nos brindará un mejor rendimiento en base a sus tiempos y costos.

Con la siguiente fórmula se puede calcular el tiempo total de ejecución:

$$T = \frac{T_x \; Q}{q \; N_{ins}} \qquad (\alpha)$$

En donde T_x es el tiempo de ejecución que se obtiene utilizando la línea de regresión de cada instancia, que se mencionó en la sección anterior. La variable Q hace referencia al intervalo entero, es decir el número de ficheros total que se desea ejecutar y la variable q indica cuantos ficheros se ejecutarán por tarea. Por último, N_{ins} es el número de instancias que se desea utilizar para la ejecución.

Además de calcular el tiempo total de ejecución, también se debe calcular el costo de ejecución y para ello se define la siguiente fórmula:

$$C = C_h \; N_{ins} \lceil T \rceil \qquad (\beta)$$

En donde C_h es el precio por hora de uso de la instancia y depende de la región en donde se utilice, estos datos se pueden observar en las Tablas 1 y 2. La variable N_{ins}, es el número de instancias que se utilizarán para la ejecución y por último, T es el tiempo total de ejecución que se obtiene de la fórmula (α), pero redondeado a un número entero y está expresado en horas.

Para determinar la instancia EC2 con la mejor configuración, se utilizará la métrica del Coste/Rendimiento [38] y de esa manera seleccionar la instancia que obtenga el menor valor.

La métrica C/R, se calcula multiplicando el Costo (C) por el Tiempo total de ejecución (T) de la instancia que se desea evaluar, estos datos se obtienen utilizando las fórmulas (α) y (β).

La fórmula general del modelo, es expresada de la siguiente manera:

$$Min\,(C/R) = Min\,(C\,T) = Min\,\left(\frac{C_h\,T_x\,Q}{q}\left\lceil\frac{T_x\,Q}{q\,N_{ins}}\right\rceil\right)$$

7. Resultados Analíticos

Con las fórmulas de la Sección 6.2, se han obtenido los resultados analíticos de los tiempos de ejecución y de los valores de Coste/Rendimiento de las instancias EC2 que se mencionan en las Tablas 1 y 2, tanto para el caso de Amazon EC2 y de Amazon EC2 con Docker.

Para obtener los mencionados resultados analíticos, se establecieron intervalos de ejecución de 1, 2, 3, 4, 5 y 6 millones de ficheros.

7.1. En Amazon EC2

En esta sección se muestran los resultados analíticos de los tiempos de ejecución y de los valores de Coste/Rendimiento de las instancias de Amazon EC2.

7.1.1. Tiempo

Los resultados analíticos de los tiempos de ejecución, se obtuvieron utilizando la fórmula (α) de la Sección 6.2, con los 6 intervalos de ejecución establecidos. Estos tiempos se encuentran en horas y están redondeados según el método de pago de las instancias EC2.

Región EU (Frankfurt)

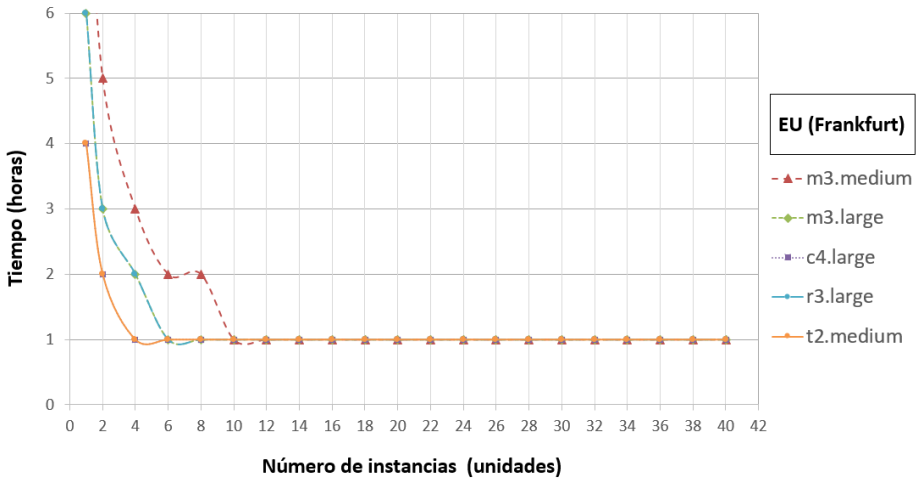

Figura 7.1. *Tiempos al procesar 1 millón de ficheros en la región EU (Frankfurt) de Amazon EC2.*

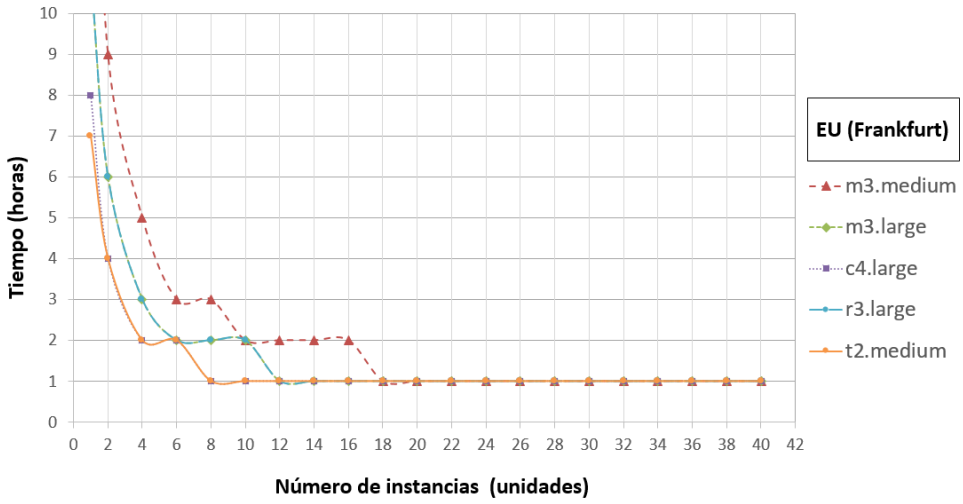

Figura 7.2. *Tiempos al procesar 2 millones de ficheros en la región EU (Frankfurt) de Amazon EC2.*

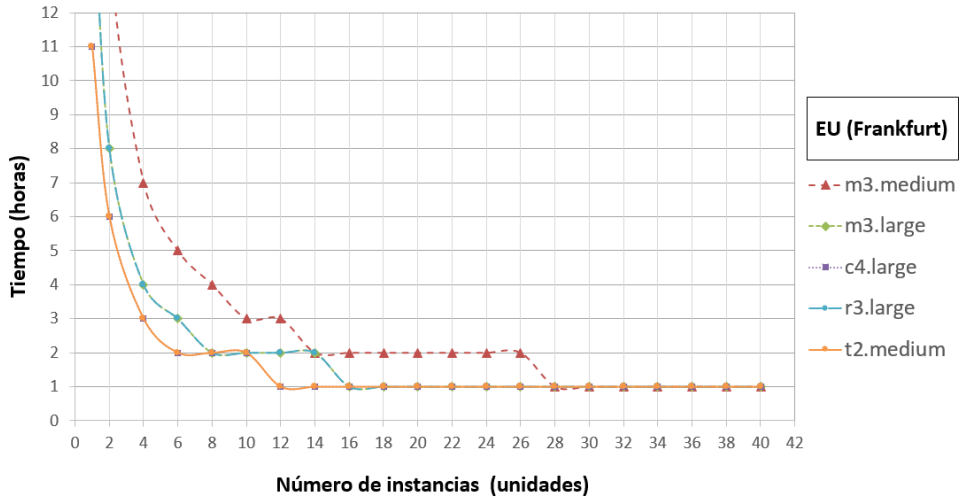

Figura 7.3. *Tiempos al procesar 3 millones de ficheros en la región EU (Frankfurt) de Amazon EC2.*

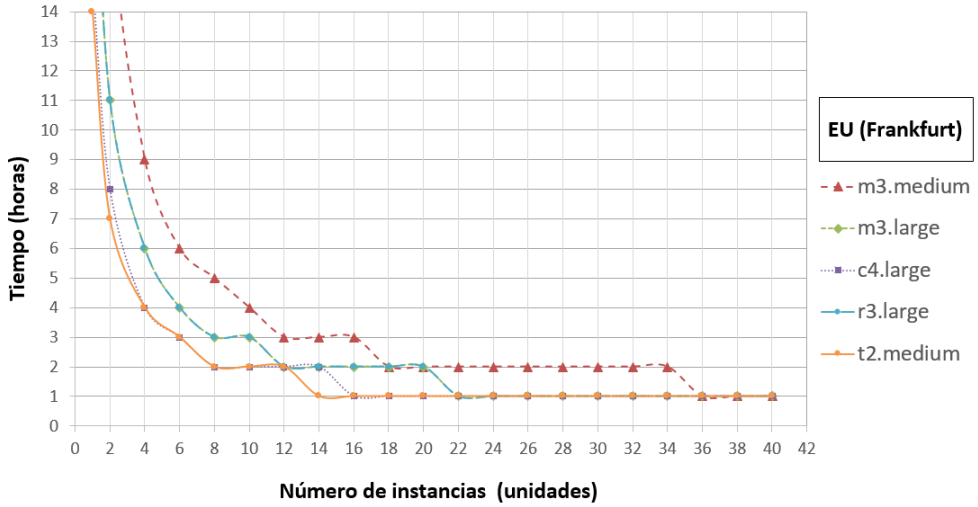

Figura 7.4. *Tiempos al procesar 4 millones de ficheros en la región EU (Frankfurt) de Amazon EC2.*

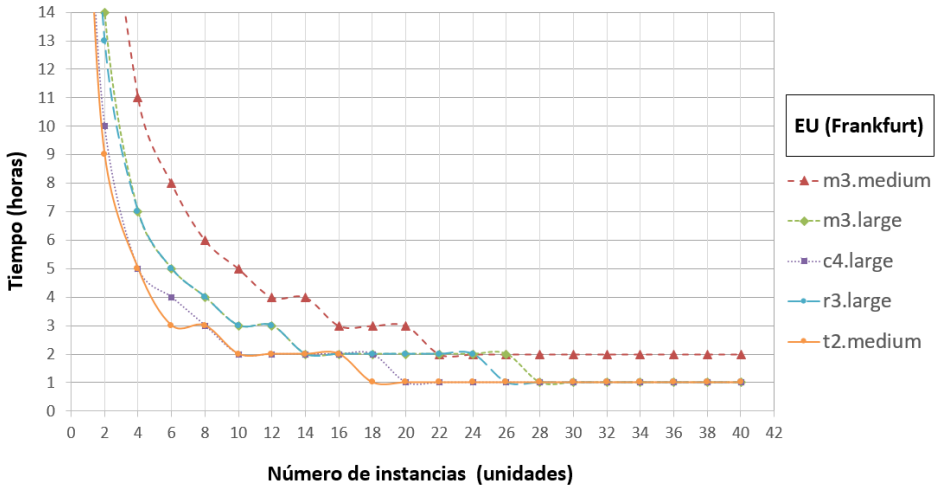

Figura 7.5. *Tiempos al procesar 5 millones de ficheros en la región EU (Frankfurt) de Amazon EC2.*

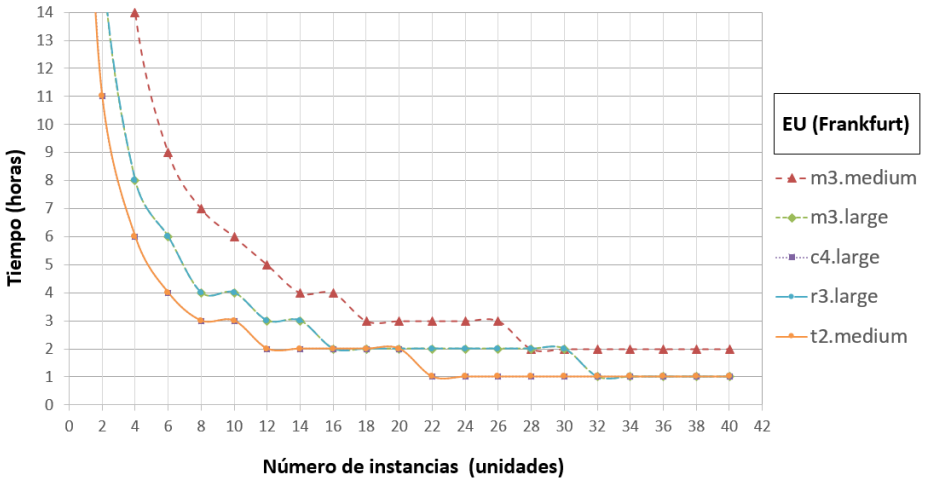

Figura 7.6. *Tiempos al procesar 6 millones de ficheros en la región EU (Frankfurt) de Amazon EC2.*

Región EE.UU. (Norte de Virginia)

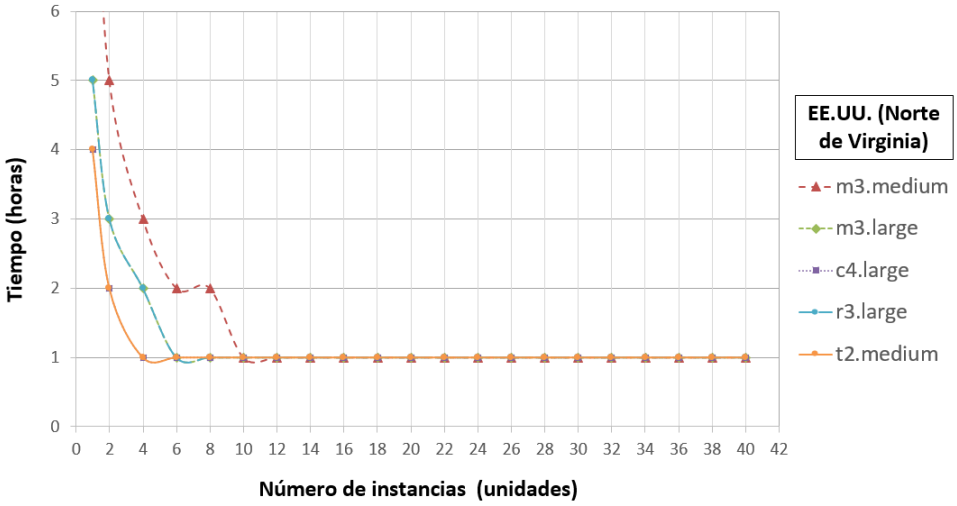

Figura 7.7. *Tiempos al procesar 1 millón de ficheros en región EE.UU (Norte de Virginia) de Amazon EC2.*

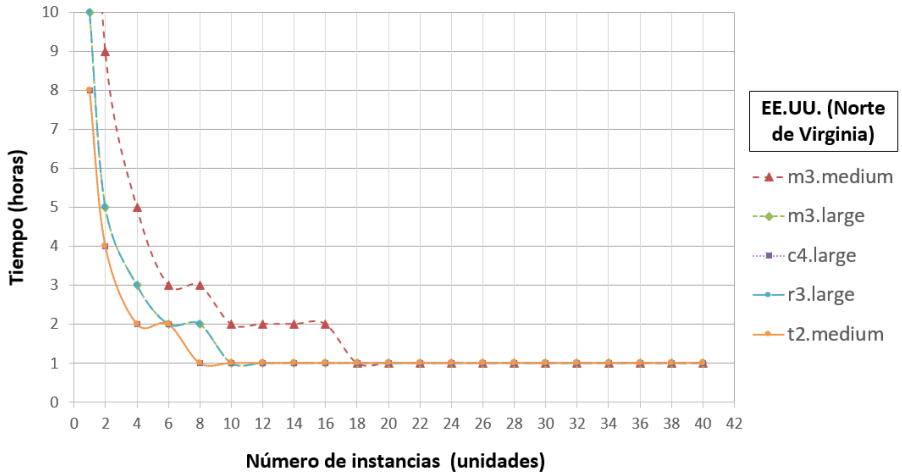

Figura 7.8. *Tiempos al procesar 2 millones de ficheros en la región EE.UU (Norte de Virginia) de Amazon EC2.*

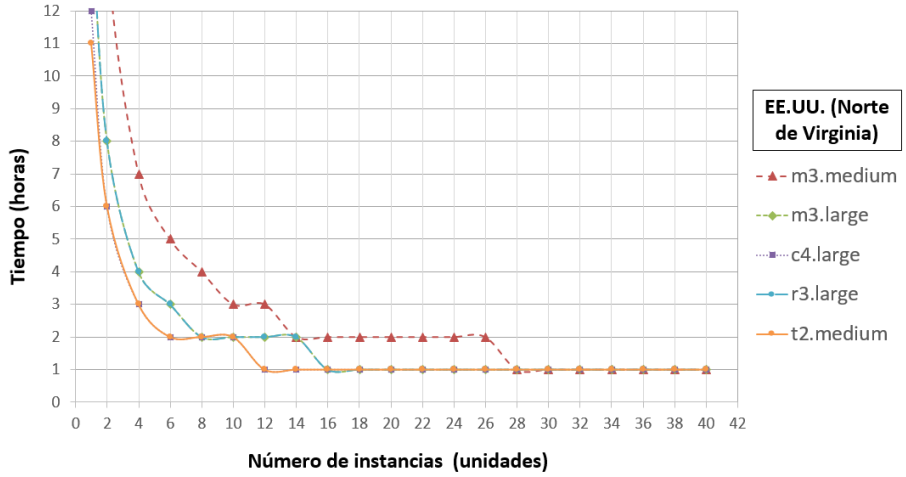

Figura 7.9. *Tiempos al procesar 3 millones de ficheros en la región EE.UU (Norte de Virginia) de Amazon EC2.*

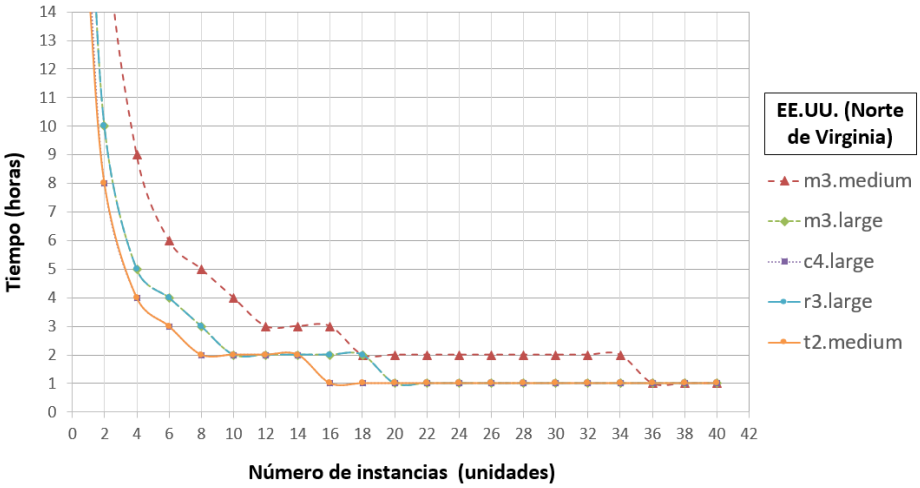

Figura 7.10. *Tiempos al procesar 4 millones de ficheros en la región EE.UU (Norte de Virginia) de Amazon EC2.*

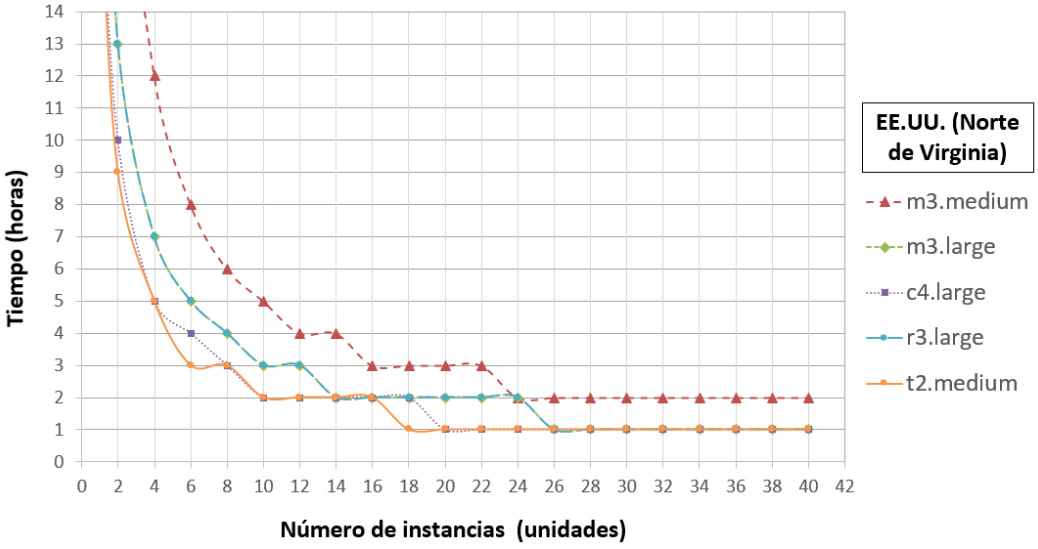

Figura 7.11. *Tiempos al procesar 5 millones de ficheros en la región EE.UU (Norte de Virginia) de Amazon EC2.*

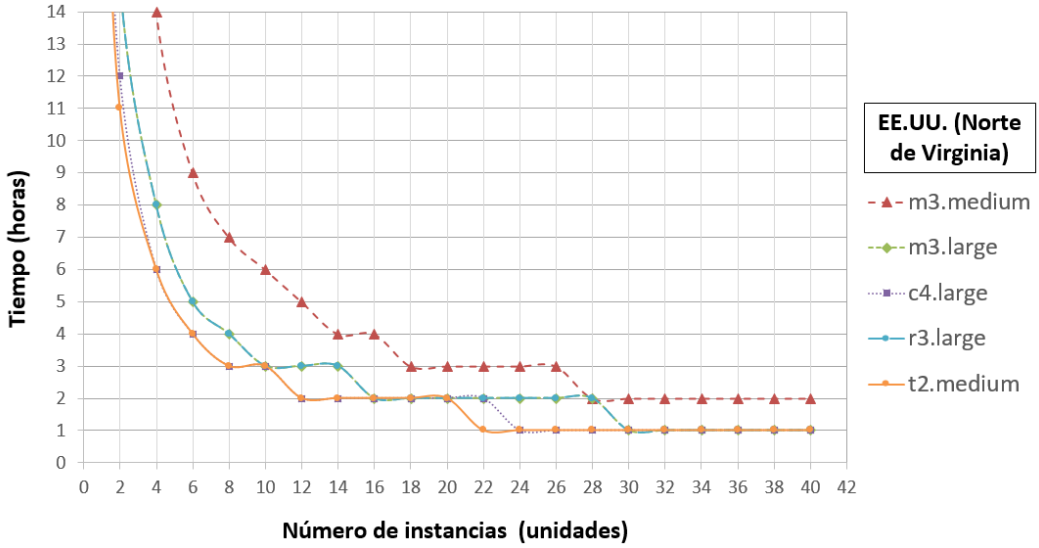

Figura 7.12. *Tiempos al procesar 6 millones de ficheros en la región EE.UU (Norte de Virginia) de Amazon EC2.*

Analizando las Figuras 7.1, 7.2, 7.3, 7.4, 7.5, 7.6, 7.7, 7.8, 7.9, 7.10, 7.11 y 7.12, podemos ver que los tiempos de ejecución en ambas regiones va disminuyendo mientras se incrementa el número de instancias utilizadas para la ejecución de la aplicación ETL.

También se puede deducir que la instancia m3.medium, obtuvo los peores tiempos de ejecución y esto se debe a que las capacidades que posee son relativamente bajas.

Las instancias r3.large y m3.large casi obtienen los mismos tiempos de ejecución y en ambas regiones, esto se debe a que sus características son similares.

Las instancias c4.large y t2.medium, obtienen los mejores tiempos de ejecución en ambas regiones, debido a que poseen las mejores características de cómputo de todas las instancias EC2 utilizadas. Además la similitud en sus tiempos de ejecución, se debe a que sus características son similares.

Los tiempos de ejecución de la instancia t2.medium son los mejores, ya que tienen una pequeña ventaja sobre la instancia c4.large.

7.1.2. Coste/Rendimiento

Los resultados analíticos de los valores de Coste/Rendimiento, se obtuvieron utilizando la fórmula general del modelo de la Sección 6.2, con los 6 intervalos de ejecución establecidos.

Región EU (Frankfurt)

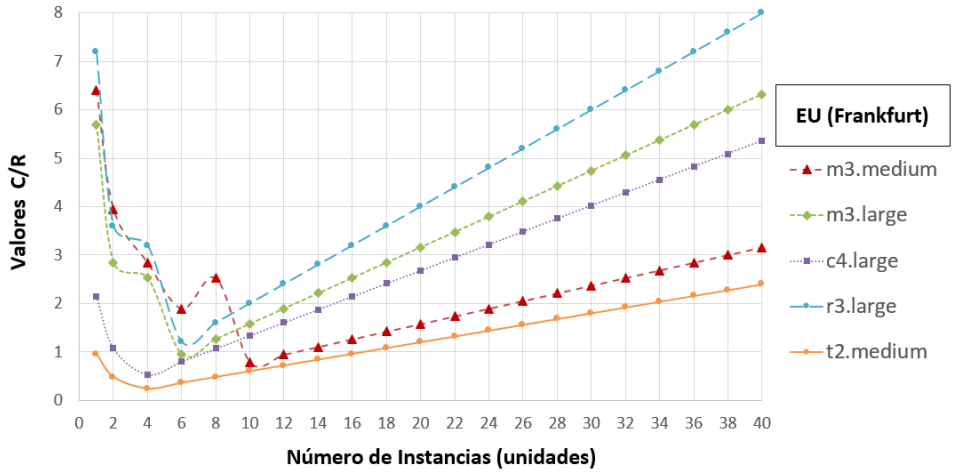

Figura 7.13. *Valores C/R al procesar 1 millón de ficheros en la región EU (Frankfurt) de Amazon EC2.*

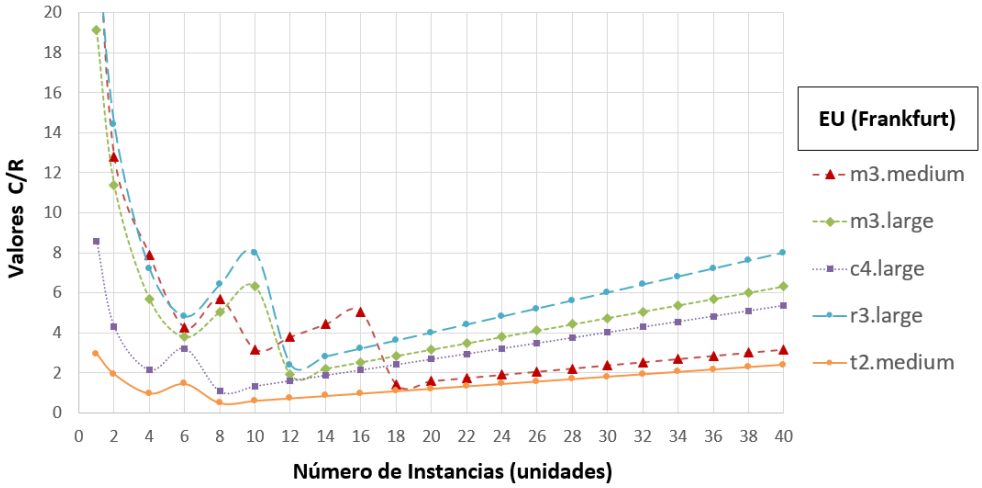

Figura 7.14. *Valores C/R al procesar 2 millones de ficheros en la región EU (Frankfurt) de Amazon EC2.*

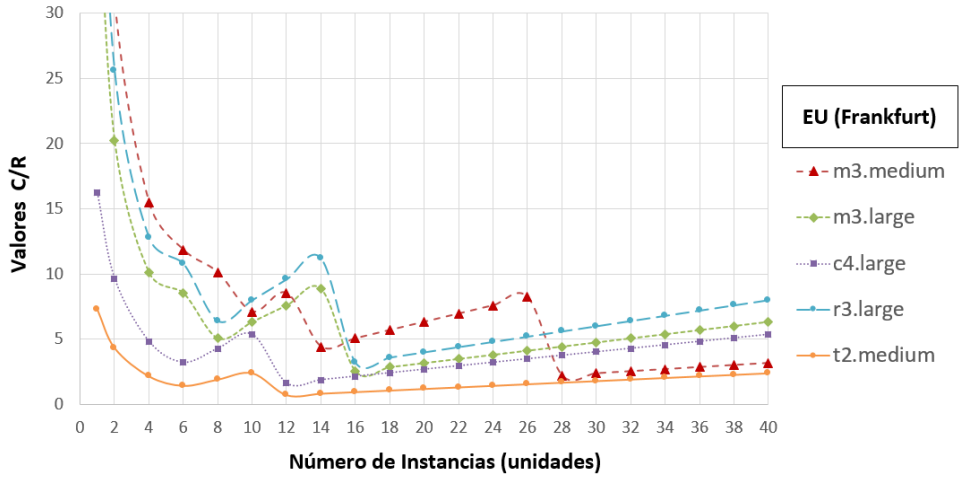

Figura 7.15. *Valores C/R al procesar 3 millones de ficheros en la región EU (Frankfurt) de Amazon EC2.*

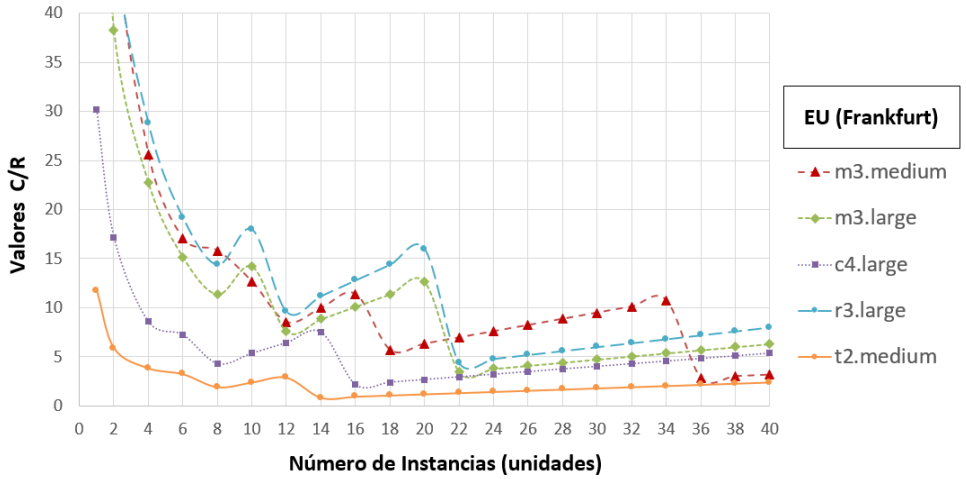

Figura 7.16. *Valores C/R al procesar 4 millones de ficheros en la región EU (Frankfurt) de Amazon EC2.*

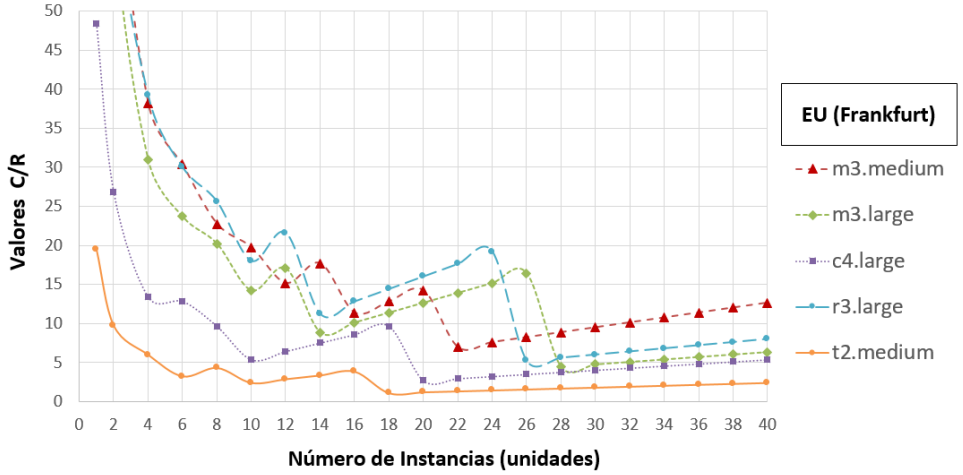

Figura 7.17. *Valores C/R al procesar 5 millones de ficheros en la región EU (Frankfurt) de Amazon EC2.*

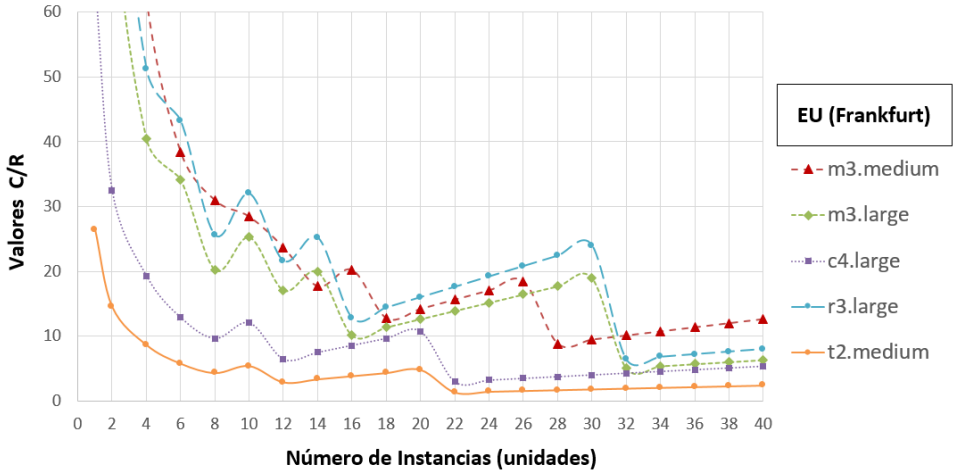

Figura 7.18. *Valores C/R al procesar 6 millones de ficheros en la región EU (Frankfurt) de Amazon EC2.*

Región EE.UU. (Norte de Virginia)

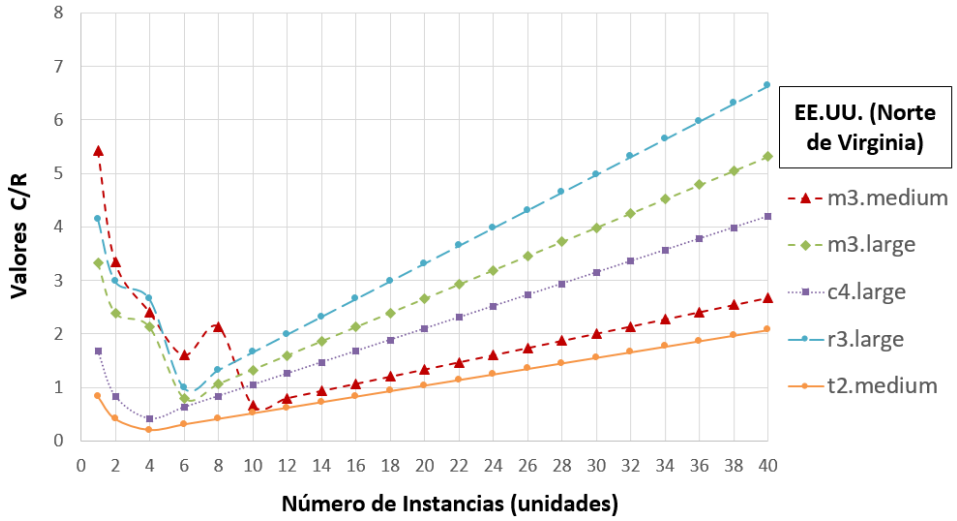

Figura 7.19. *Valores C/R al procesar 1 millón de ficheros en la región EE.UU (Norte de Virginia) de Amazon EC2.*

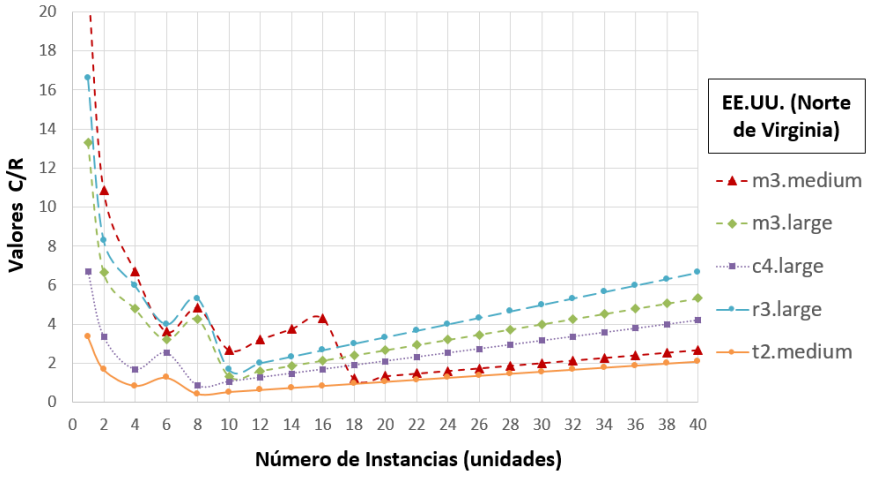

Figura 7.20. *Valores C/R al procesar 2 millones de ficheros en la región EE.UU (Norte de Virginia) de Amazon EC2.*

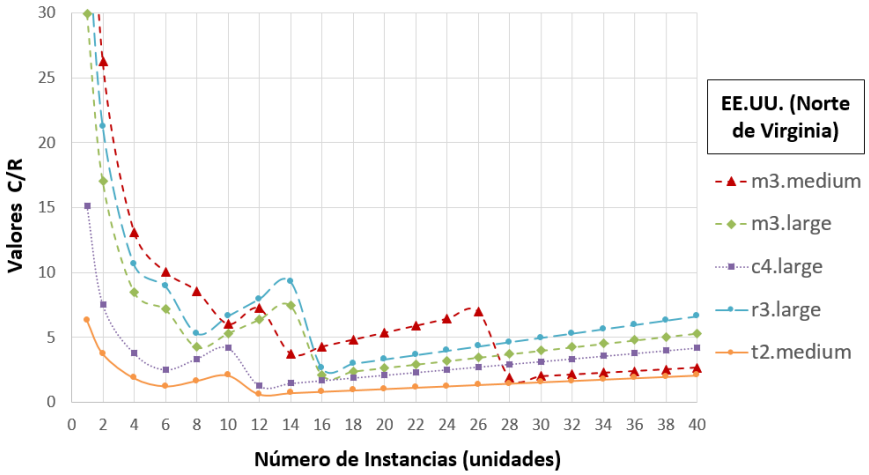

Figura 7.21. *Valores C/R al procesar 3 millones de ficheros en la región EE.UU (Norte de Virginia) de Amazon EC2.*

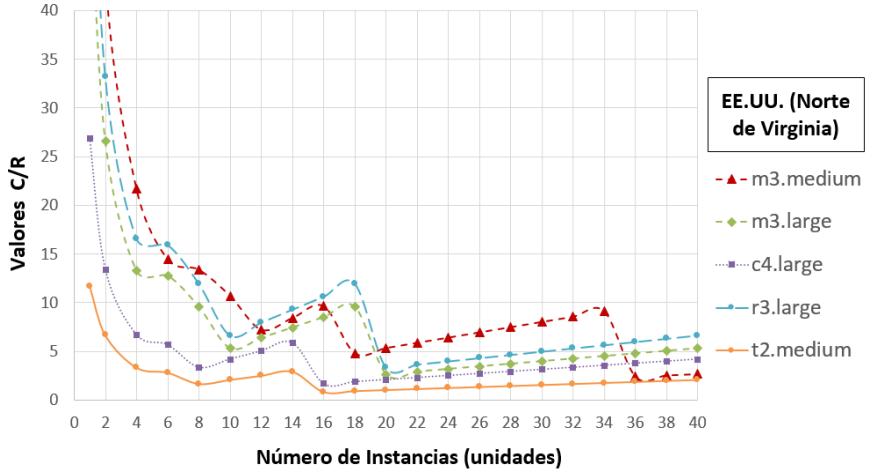

Figura 7.22. *Valores C/R al procesar 4 millones de ficheros en la región EE.UU (Norte de Virginia) de Amazon EC2.*

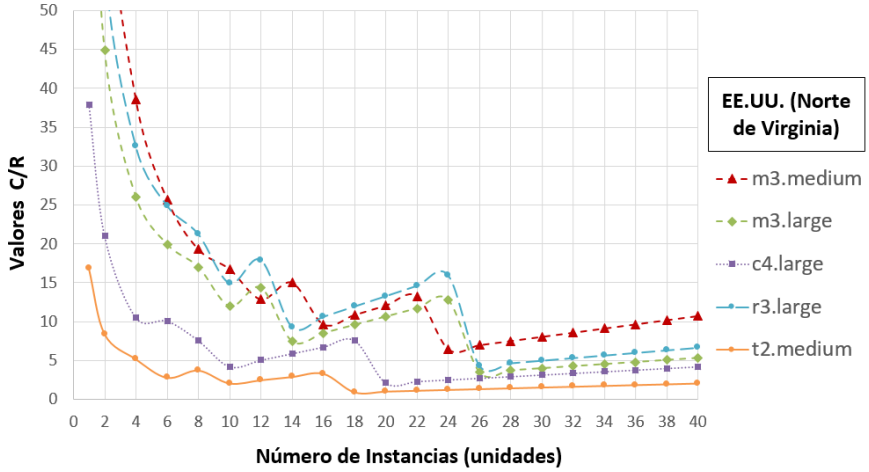

Figura 7.23. *Valores C/R al procesar 5 millones de ficheros en la región EE.UU (Norte de Virginia) de Amazon EC2.*

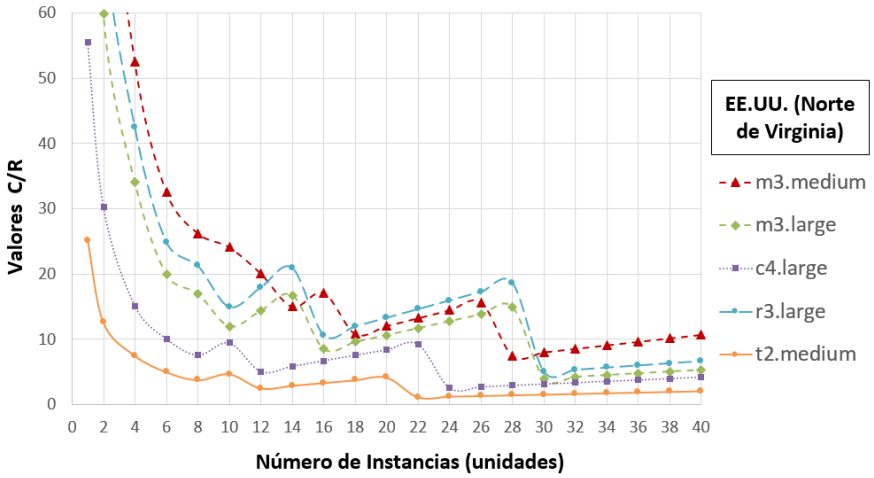

Figura 7.24. *Valores C/R al procesar 6 millones de ficheros en la región EE.UU (Norte de Virginia) de Amazon EC2.*

Analizando las Figuras 7.13, 7.14, 7.15, 7.16, 7.17, 7.18, 7.19, 7.20, 7.21, 7.22, 7.23 y 7.24, podemos ver que los valores de Coste/Rendimiento aumentan y disminuyen, con lo cual se forman unas crestas.

Estas crestas en algunos casos son pequeñas y en otros grandes, pero para entender cómo se originan, analizaremos los resultados de C/R obtenidos en la Figura 7.21. Para complementar este análisis, se revisarán los tiempos de ejecución obtenidos en la Figura 7.9.

Para la instancia r3.large, al utilizar 12 instancias se obtiene un valor C/R de 9,6, en un tiempo de ejecución de 2 horas. Cuando se utilizan 14 instancias, el valor C/R aumenta a 11,2, en un tiempo de ejecución de 2 horas, pero al utilizar 16 instancias el valor C/R disminuye a 3,2, en un tiempo de ejecución de 1 hora.

Entonces podemos deducir que cuando el tiempo de ejecución se mantiene en el mismo valor, los valores C/R aumentan si el número de instancias aumenta. En el momento en que el tiempo de ejecución disminuye, el valor C/R también disminuye, así se van formando las crestas en las gráficas. También podemos observar que todas las instancias después de haber obtenido su valor C/R mínimo y con el aumento del número de instancias, el valor C/R aumente constantemente.

Por último, en ambas regiones la instancia t2.medium obtiene el menor valor C/R, además de utilizar el menor número de instancias para obtener ese valor.

7.2. En Amazon EC2 con Docker

En esta sección se muestran los resultados analíticos de los tiempos de ejecución y de los valores de Coste/Rendimiento de las instancias de Amazon EC2 con Docker.

7.2.1. Tiempo

De igual manera que en la Sección 7.1.1, los resultados analíticos de los tiempos de ejecución, se obtuvieron utilizando la fórmula (α) de la Sección 6.2, con los 6 intervalos de ejecución establecidos. Debemos tener en cuenta que estos tiempos se encuentran en horas y están redondeados según el método de pago de las instancias EC2.

Región EU (Frankfurt)

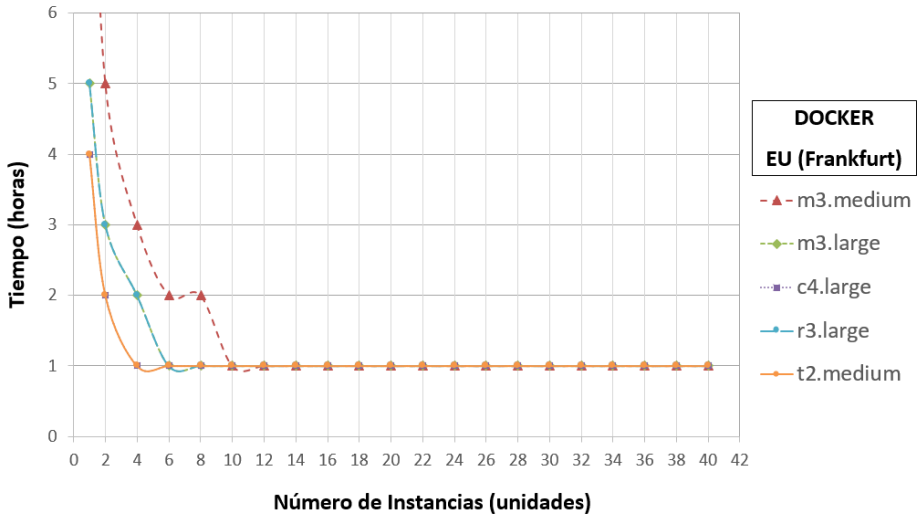

Figura 7.25. *Tiempos al procesar 1 millón de ficheros en la región EU (Frankfurt) de Amazon EC2 con Docker.*

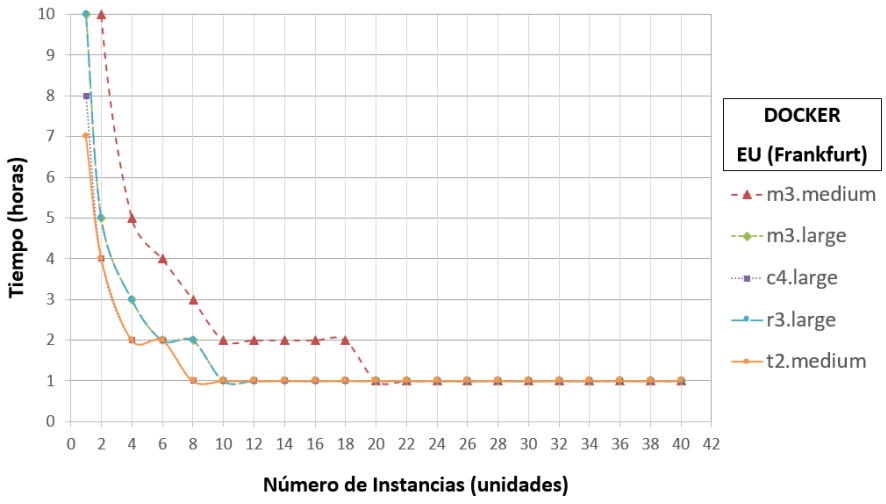

Figura 7.26. *Tiempos al procesar 2 millones de ficheros en la región EU (Frankfurt) de Amazon EC2 con Docker.*

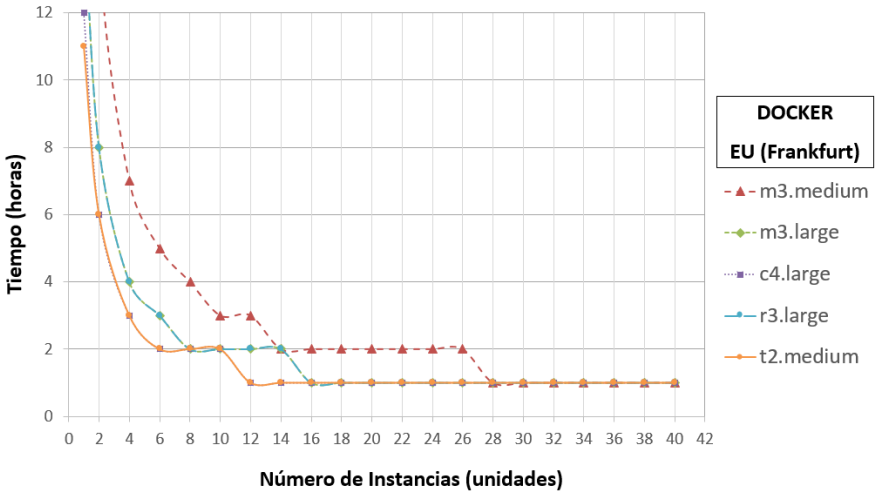

Figura 7.27. *Tiempos al procesar 3 millones de ficheros en la región EU (Frankfurt) de Amazon EC2 con Docker.*

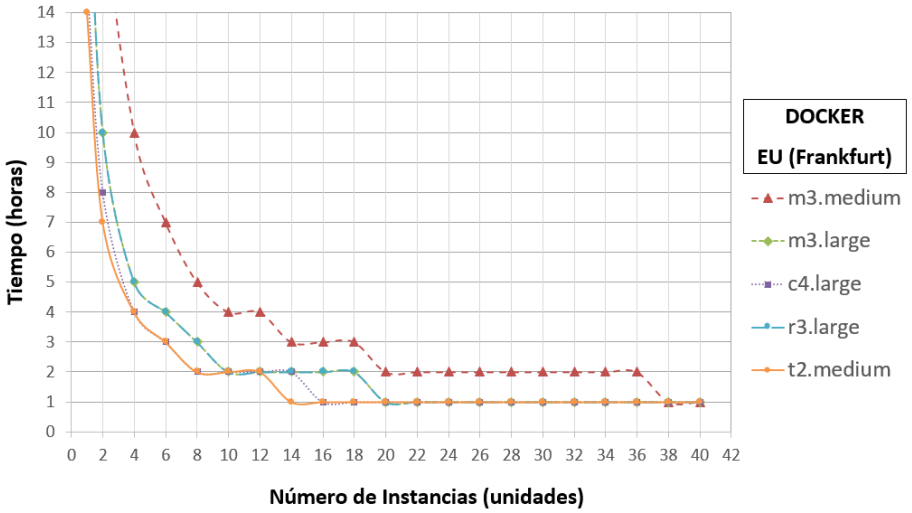

Figura 7.28. *Tiempos al procesar 4 millones de ficheros en la región EU (Frankfurt) de Amazon EC2 con Docker.*

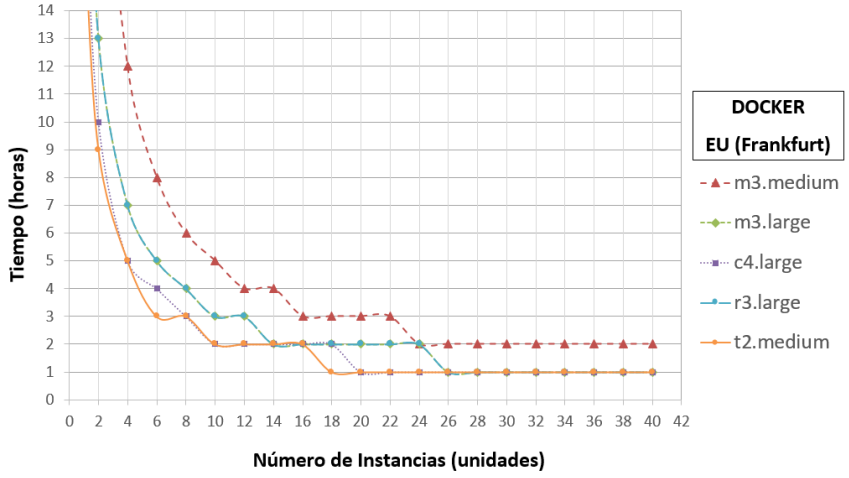

Figura 7.29. *Tiempos al procesar 5 millones de ficheros en la región EU (Frankfurt) de Amazon EC2 con Docker.*

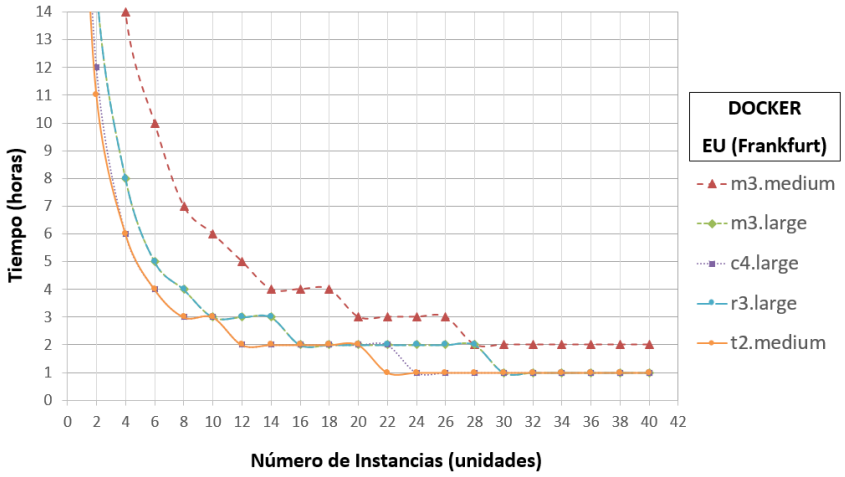

Figura 7.30. *Tiempos al procesar 6 millones de ficheros en la región EU (Frankfurt) de Amazon EC2 con Docker.*

Región EE.UU. (Norte de Virginia)

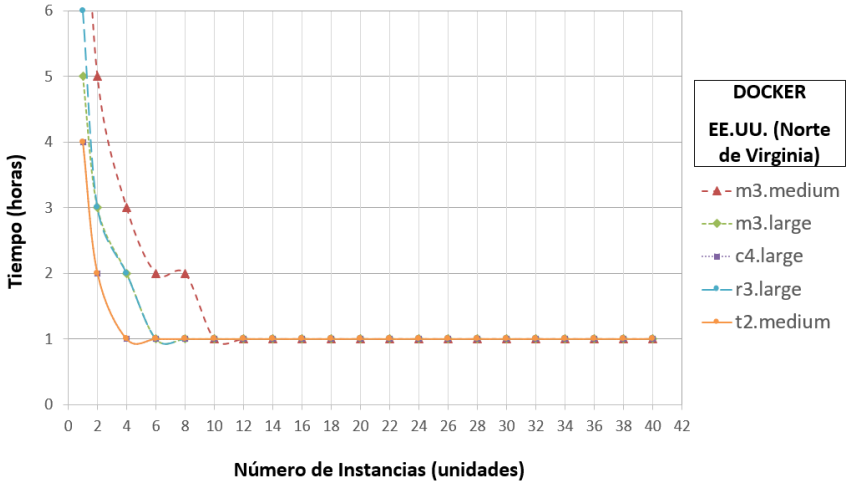

Figura 7.31. *Tiempos al procesar 1 millón de ficheros en la región EE.UU (Norte de Virginia) de Amazon EC2 con Docker.*

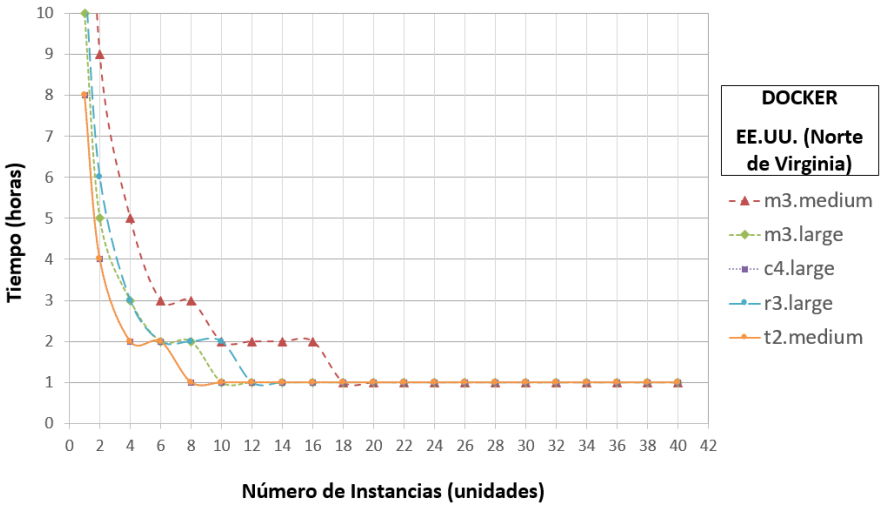

Figura 7.32. *Tiempos al procesar 2 millones de ficheros en la región EE.UU (Norte de Virginia) de Amazon EC2 con Docker.*

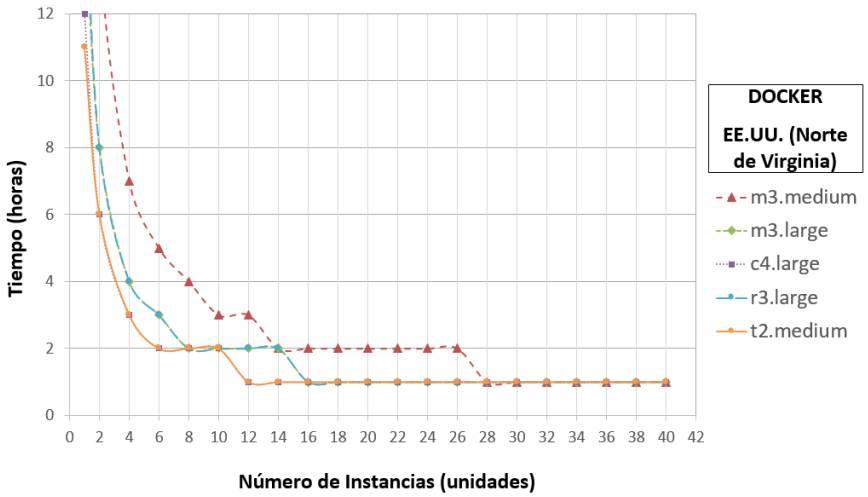

Figura 7.33. *Tiempos al procesar 3 millones de ficheros en la región EE.UU (Norte de Virginia) de Amazon EC2 con Docker.*

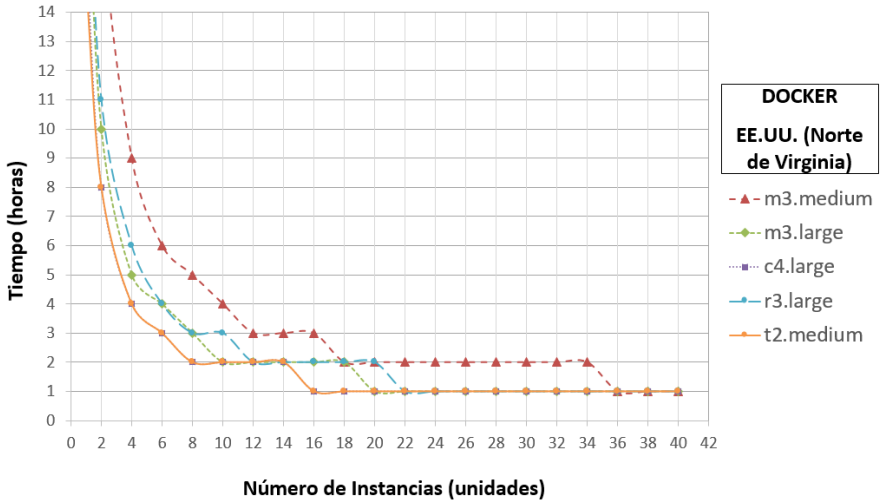

Figura 7.34. *Tiempos al procesar 4 millones de ficheros en la región EE.UU (Norte de Virginia) de Amazon EC2 con Docker.*

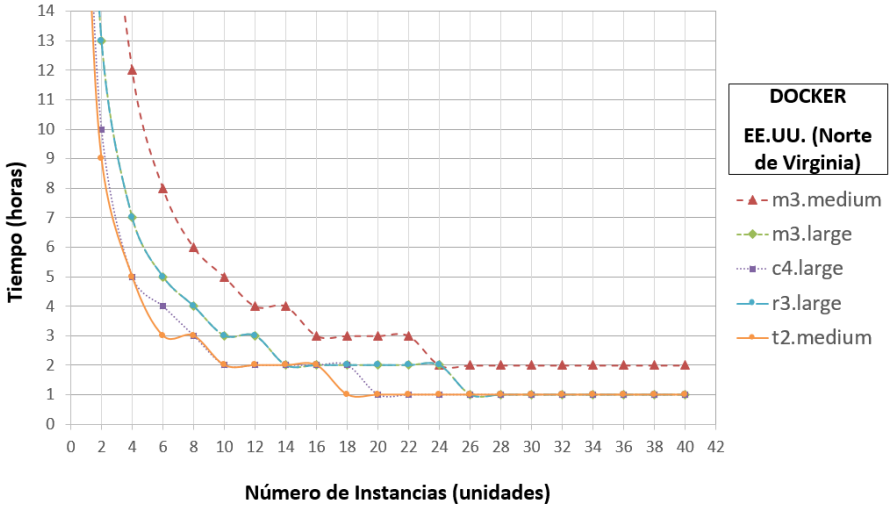

Figura 7.35. *Tiempos al procesar 5 millones de ficheros en la región EE.UU (Norte de Virginia) de Amazon EC2 con Docker.*

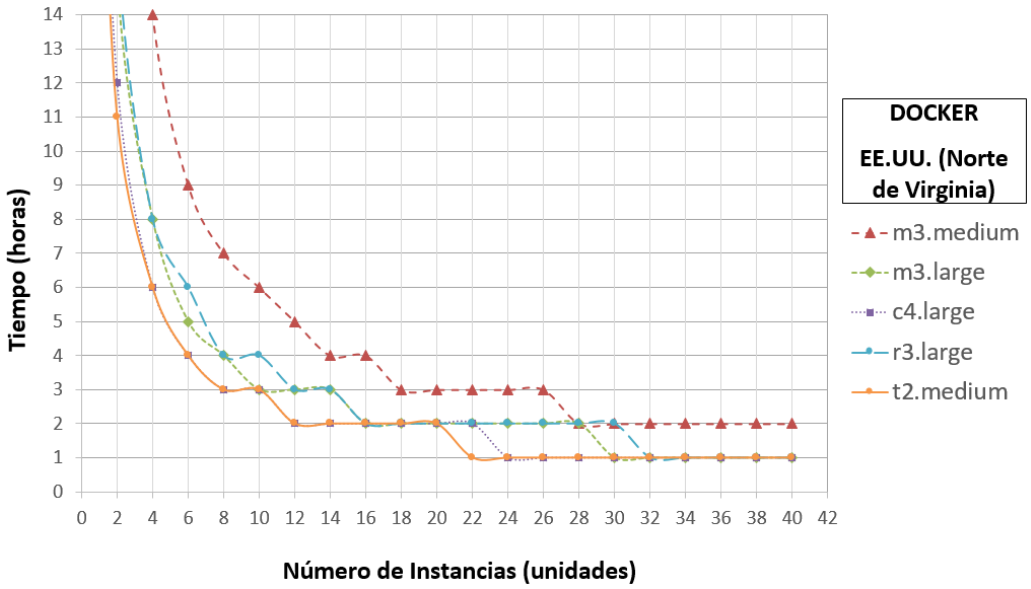

Figura 7.36. *Tiempos al procesar 6 millones de ficheros en la región EE.UU (Norte de Virginia) de Amazon EC2 con Docker.*

Analizando las Figuras 7.25, 7.26, 7.27, 7.28, 7.29, 7.30, 7.31, 7.32, 7.33, 7.34, 7.35 y 7.36, al igual que las gráficas de la Sección 7.1.1, podemos ver que los tiempos de ejecución en ambas regiones va disminuyendo, mientras que el número de instancias utilizadas para la ejecución de la aplicación ETL se incrementa.

Además, se puede observar que nuevamente la instancia m3.medium ofrece el peor tiempo de ejecución. Las instancias m3.large y r3.large, obtienen resultados casi iguales en ambas regiones, debido a la similitud de sus características.

Las instancias c4.large y t2.medium, obtienen los mejores tiempos de ejecución en ambas regiones. Los tiempos de ambas regiones son casi iguales y esto se debe a que sus características de cómputo son similares.

En todas las gráficas podemos observar que los tiempos de ejecución de la instancia t2.medium son los mejores.

7.2.2. Coste/Rendimiento

Los resultados analíticos de los valores de Coste/Rendimiento, se obtuvieron utilizando la fórmula general del modelo de la Sección 6.2, con los 6 intervalos de ejecución establecidos.

Región EU (Frankfurt)

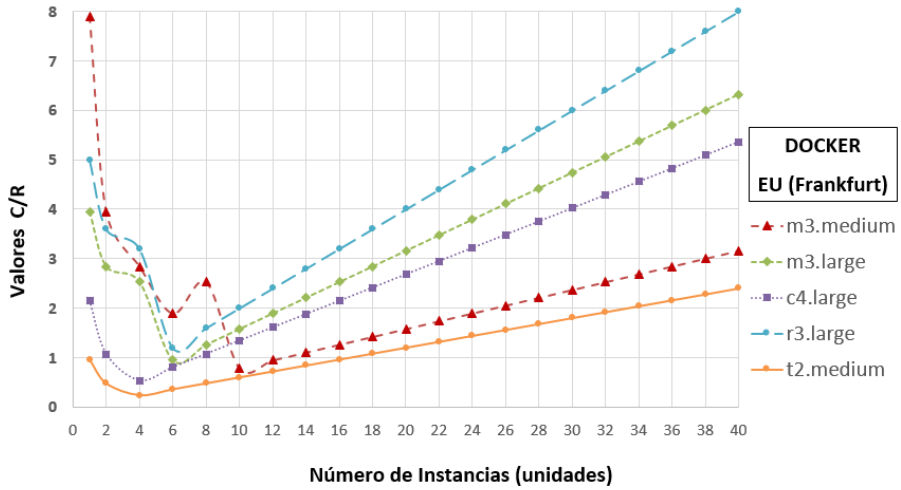

Figura 7.37. *Valores C/R al procesar 1 millón de ficheros en la región EU (Frankfurt) de Amazon EC2 con Docker.*

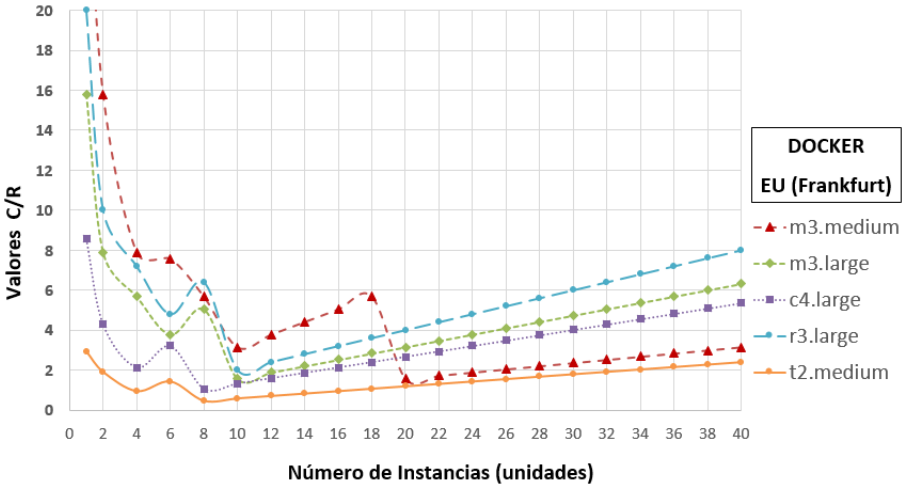

Figura 7.38. *Valores C/R al procesar 2 millones de ficheros en la región EU (Frankfurt) de Amazon EC2 con Docker.*

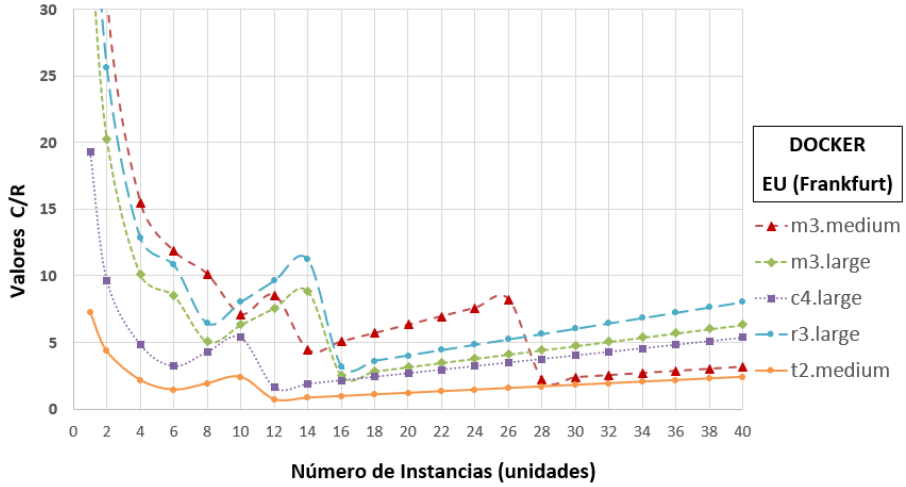

Figura 7.39. *Valores C/R al procesar 3 millones de ficheros en la región EU (Frankfurt) de Amazon EC2 con Docker.*

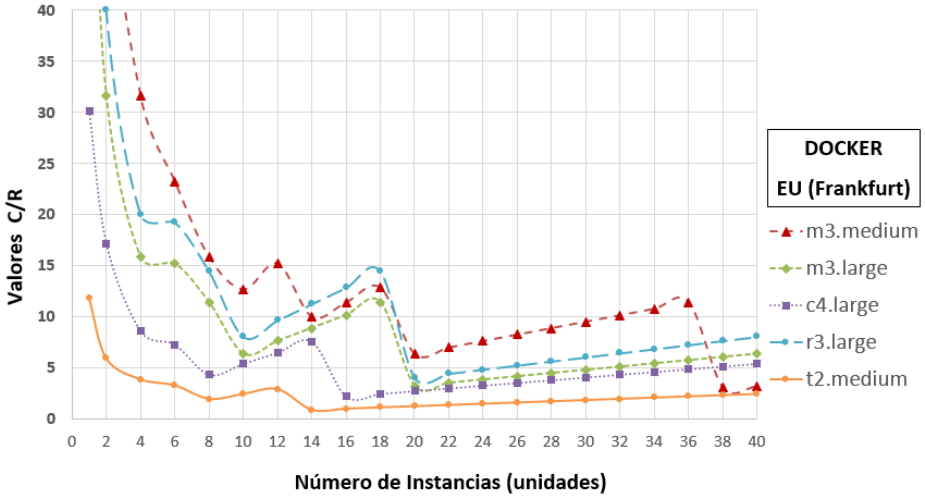

Figura 7.40. *Valores C/R al procesar 4 millones de ficheros en la región EU (Frankfurt) de Amazon EC2 con Docker.*

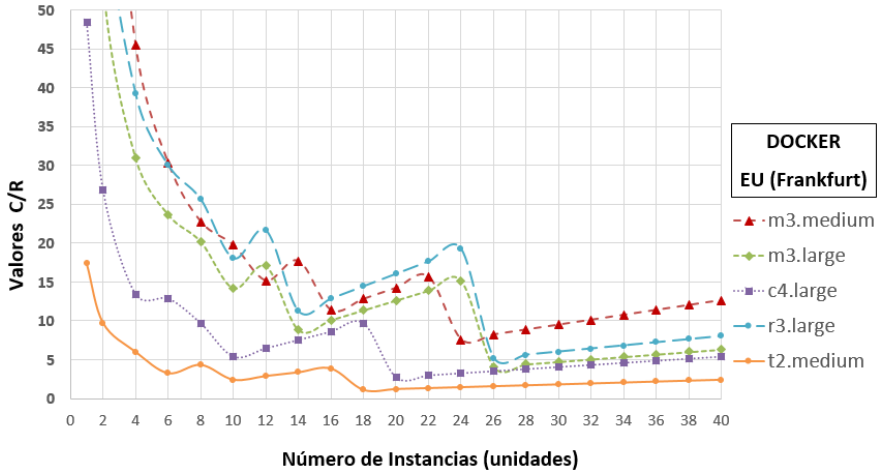

Figura 7.41. *Valores C/R al procesar 5 millones de ficheros en la región EU (Frankfurt) de Amazon EC2 con Docker.*

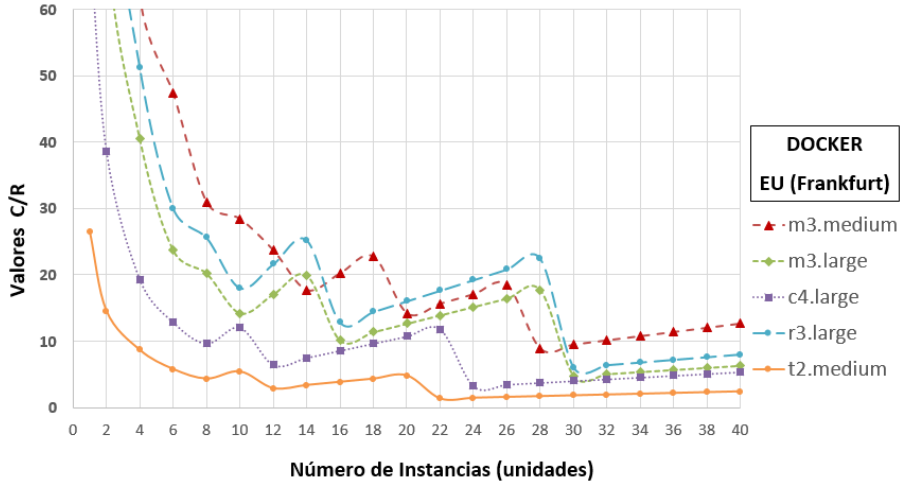

Figura 7.42. *Valores C/R al procesar 6 millones de ficheros en la región EU (Frankfurt) de Amazon EC2 con Docker.*

Región EE.UU. (Norte de Virginia)

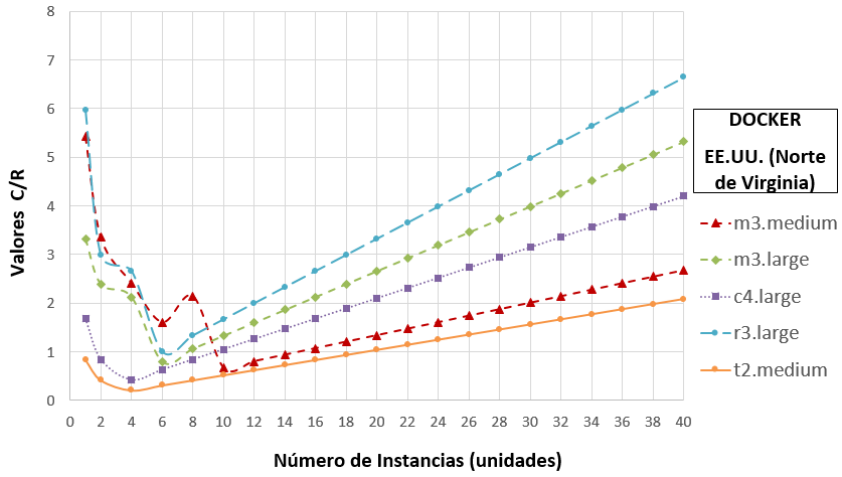

Figura 7.43. *Valores C/R al procesar 1 millón de ficheros en la región EE.UU (Norte de Virginia) de Amazon EC2 con Docker.*

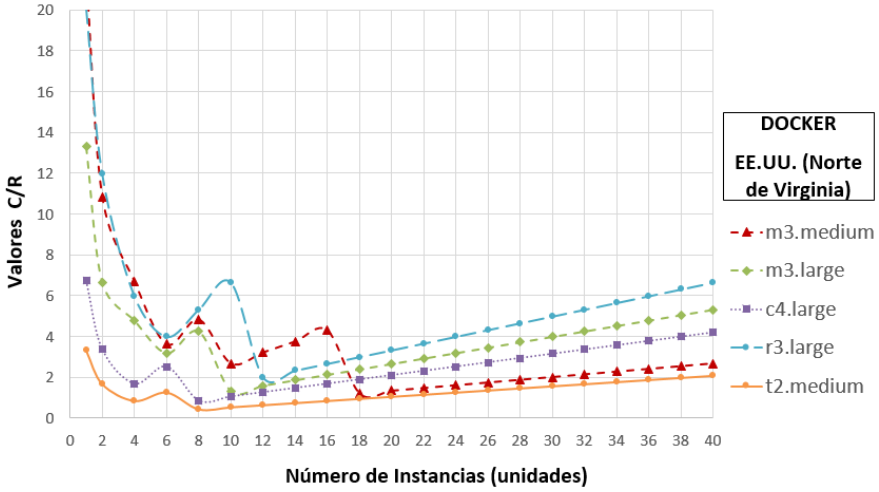

Figura 7.44. *Valores C/R al procesar 2 millones de ficheros en la región EE.UU (Norte de Virginia) de Amazon EC2 con Docker.*

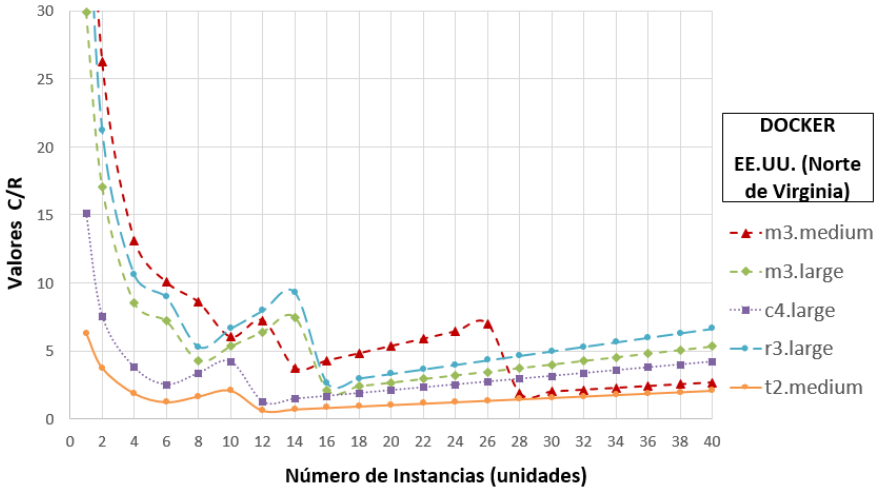

Figura 7.45. *Valores C/R al procesar 3 millones de ficheros en la región EE.UU (Norte de Virginia) de Amazon EC2 con Docker.*

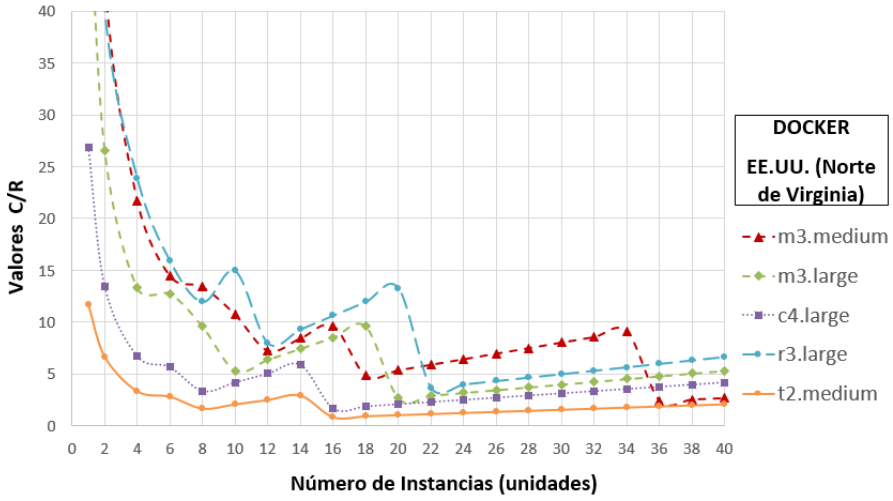

Figura 7.46. *Valores C/R al procesar 4 millones de ficheros en la región EE.UU (Norte de Virginia) de Amazon EC2 con Docker.*

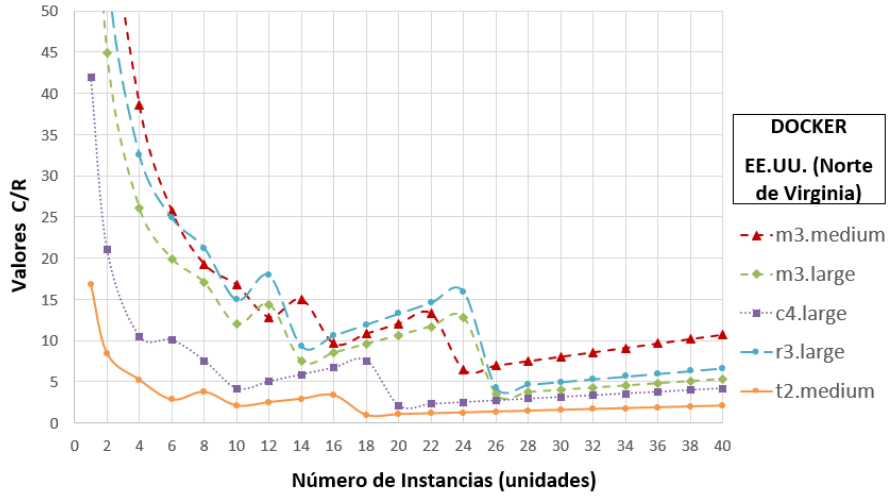

Figura 7.47. *Valores C/R al procesar 5 millones de ficheros en la región EE.UU (Norte de Virginia) de Amazon EC2 con Docker.*

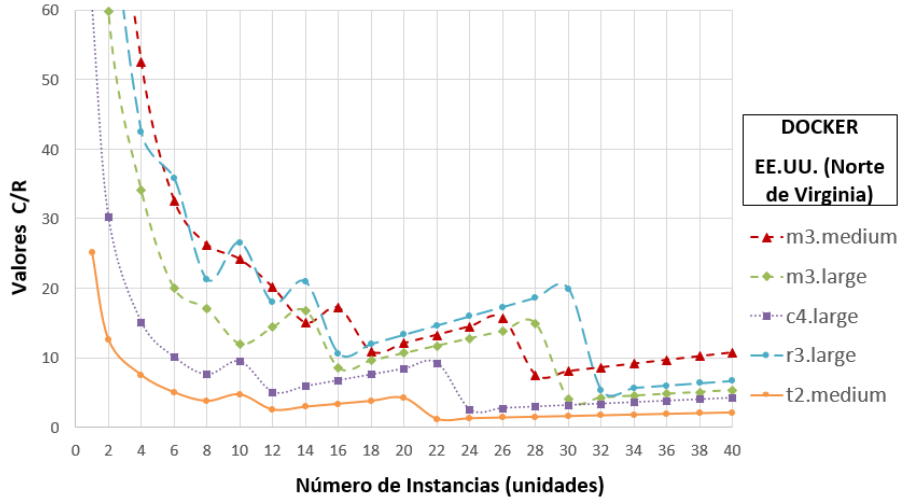

Figura 7.48. *Valores C/R al procesar 6 millones de ficheros en la región EE.UU (Norte de Virginia) de Amazon EC2 con Docker.*

Al igual que en la Sección 7.1.2, en las Figuras 7.37, 7.38,7.39, 7.40, 7.41, 7.42, 7.43, 7.44, 7.45, 7.46, 7.47 y 7.48, podemos observar que los valores de Coste/Rendimiento obtenidos aumentan y disminuyen, formando crestas.

Con los resultados de los valores C/R obtenidos en la Figura 7.40 y los tiempos de ejecución obtenidos en la Figura 7.28, se analizaron como se forman las crestas de los valores C/R.

Para la instancia c4.large, al utilizar 10 instancias se obtiene un valor C/R de 5.36, en un tiempo de ejecución de 2 horas. Cuando se utilizan 12 instancias, el valor C/R aumenta a 6,43, en un tiempo de ejecución de 2 horas y utilizando 14 instancias, el valor C/R aumenta a 7,50, en un tiempo de ejecución de 2 horas. Podemos deducir que mientras el tiempo de ejecución se mantiene, el valor C/R aumenta si es que el número de instancias aumenta.

En el momento en que se utilizan 16 instancias, el valor C/R disminuye a 2,14, con un tiempo de ejecución de 1 hora. Cuando se utilizan 18 instancias el valor C/R aumenta a 2,41, en un tiempo de ejecución de 1 hora. Al continuar el aumento de instancias, el tiempo de ejecución se mantiene en una hora y el valor C/R aumenta constantemente.

Para todas las instancias, después de alcanzar su valor C/R mínimo, los siguientes valores C/R se mantienen en constante aumento y ya no se forman más crestas.

Por último, en ambas regiones la instancia t2.medium obtiene el menor valor C/R, además de utilizar el menor número de instancias para obtener ese valor.

8. Casos de Uso

Con el análisis de los siguientes casos, teniendo en cuenta los tiempos de ejecución y a los valores de Coste/Rendimiento que se muestran en la Sección 7, podemos determinar cuál es el mejor tipo de instancia en Amazon EC2 y en Amazon EC2 con Docker.

Como se indicó en la Sección 6.2, utilizando la fórmula general del modelo debemos obtener el valor C/R mínimo para cada tipo de instancia EC2, con diferentes números de instancias.

Caso 1: Procesando 2 millones de ficheros

En Amazon EC2:

Para analizar los datos de la región EU (Frankfurt), se deben analizar las Figuras 7.2 y 7.14, y para la región EE.UU. (Norte de Virginia), las Figuras 7.8 y 7.20.

En ambas regiones, todos los tipos de instancias obtienen sus valores C/R mínimos en una hora. En las Figuras 8.1.a y 8.1.b, podemos ver los valores C/R mínimos de cada instancia EC2 y el número de instancias que se han utilizado para obtener esos valores.

Se sabe que la instancia m3.medium es la que obtiene los peores tiempos de ejecución y para llegar a su valor C/R mínimo, se debe incrementar considerablemente el número de instancias, a diferencia de las demás instancias.

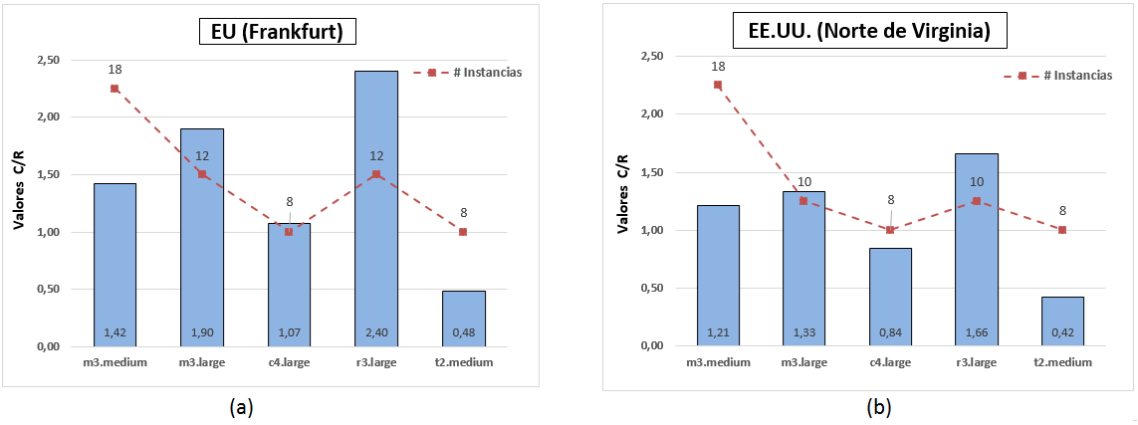

Figura 8.1. *Valores C/R al procesar 2 millones de ficheros, (a) en la Región EU (Frankfurt) y (b) en la Región EE.UU. (Norte de Virginia) de Amazon EC2.*

Para ambas regiones, la instancia t2.medium es la mejor opción para este caso, debido a que obtienen los mejores valores C/R mínimos, para la región EU (Frankfurt) un valor C/R de 0,48 y para la región EE.UU. (Norte de Virginia) un valor C/R de 0,42. Además, utilizan el mismo número de instancias para la ejecución. Después de este análisis, se puede deducir que la instancia t2.medium de la región EE.UU. (Norte de Virginia) es la mejor instancia para este caso.

En Amazon EC2 con Docker:

Para analizar los datos de la región EU (Frankfurt), se deben analizar las Figuras 7.26 y 7.38, y para la región EE.UU. (Norte de Virginia), las Figuras 7.32 y 7.44.

El tiempo utilizado por todos los tipos de instancias para obtener sus valores C/R mínimos, ha sido de una hora. En las Figuras 8.2.a y 8.2.b, podemos ver los valores C/R mínimos de cada instancia EC2 y el número de instancias que se han utilizado para obtener esos valores.

Los peores tiempos de ejecución los obtiene la instancia m3.medium y para llegar a su valor C/R mínimo, necesita utilizar un mayor número de instancias que las que utilizan las demás instancias. Por ejemplo, los tipos de instancias m3.medium y m3.large utilizan 20 y 10 instancias respectivamente, la primera utiliza casi el doble del número de instancias.

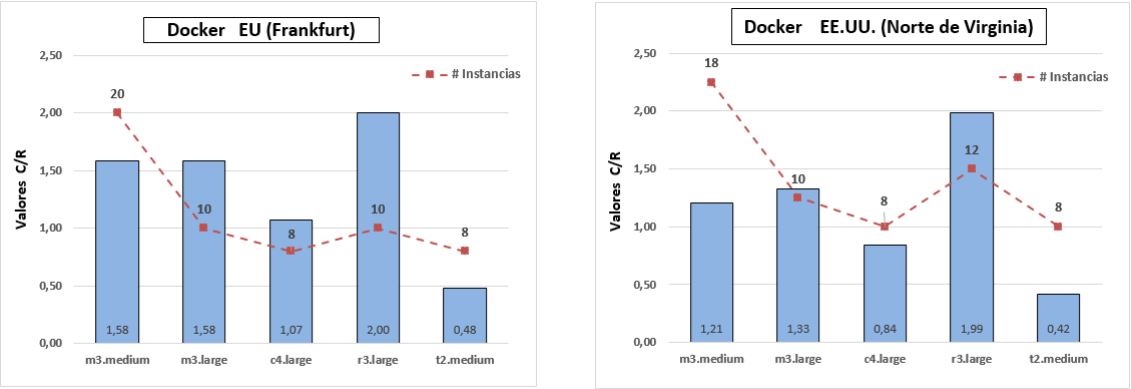

Figura 8.2. *Valores C/R al procesar 2 millones de ficheros, (a) en la Región EU (Frankfurt) y (b) en la Región EE.UU. (Norte de Virginia) de Amazon EC2 con Docker.*

La instancia t2.medium es la mejor opción para este caso, debido a que obtienen los mejores valores C/R mínimos, para la región EU (Frankfurt) un valor C/R de 0,48 y para la región EE.UU. (Norte de Virginia) un valor C/R de 0,42. Además, ambos tipos de instancias utilizan el mismo número de instancias para la ejecución, 8 en total. Después de este análisis, se puede deducir que la instancia t2.medium de la región EE.UU. (Norte de Virginia) es la mejor instancia para este caso.

Caso 2: Procesando 4 millones de ficheros

En Amazon EC2:

Para analizar los datos de la región EU (Frankfurt), se deben analizar las Figuras 7.4 y 7.16, y para la región EE.UU. (Norte de Virginia), las Figuras 7.10 y 7.22.

Podemos observar que en una hora de ejecución, todos los tipos de instancias han obtenido sus valores C/R mínimos

En las Figuras 8.3.a y 8.3.b, podemos ver los valores C/R mínimos de los tipos de instancias EC2 y el número de instancias utilizadas.

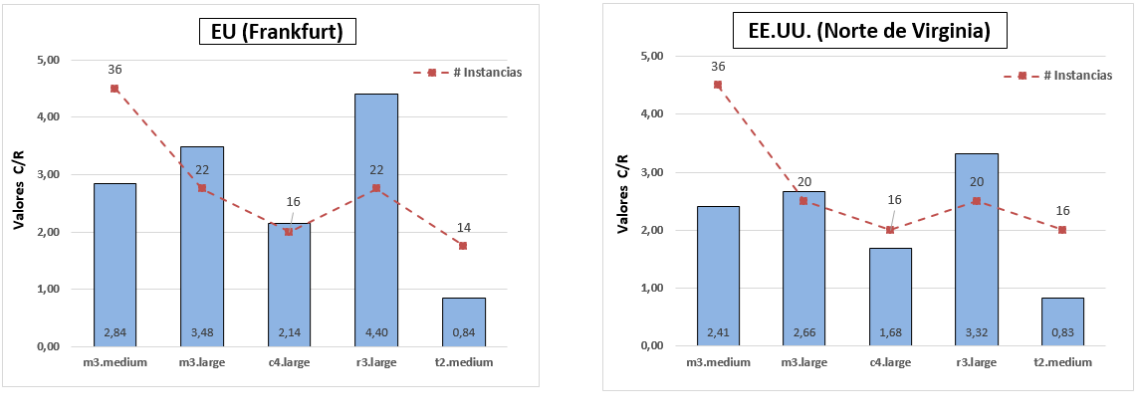

Figura 8.3. *Valores C/R al procesar 4 millones de ficheros, (a) en la Región EU (Frankfurt) y (b) en la Región EE.UU. (Norte de Virginia) de Amazon EC2.*

Podemos ver que el tipo de instancia m3.medium, en algunos casos ha utilizado el doble o más del doble de número de instancias que los demás tipos. Por ejemplo, los tipos de instancias m3.medium y t2.medium utilizan 36 y 14 instancias respectivamente, la primera ni utilizando más del doble de instancias que la segunda, obtiene un mejor valor C/R.

La instancia t2.medium es la mejor opción para este caso, debido a que obtienen los mejores valores C/R mínimos, para la región EU (Frankfurt) un valor C/R de 0,84 y para la región EE.UU. (Norte de Virginia) un valor C/R de 0,83.

Entonces podemos deducir que la mejor opción es la instancia t2.medium de la región EE.UU. (Norte de Virginia), así utilice 2 instancias más que el de la región EU (Frankfurt), su valor C/R de 0,83 es el mínimo.

En Amazon EC2 con Docker:

Para analizar los datos de la región EU (Frankfurt), se deben analizar las Figuras 7.28 y 7.40, y para la región EE.UU. (Norte de Virginia), las Figuras 7.34 y 7.46.

El tiempo utilizado por todos los tipos de instancias para obtener sus valores C/R mínimos, ha sido de una hora. En las Figuras 8.4.a y 8.4.b, podemos observar los valores C/R mínimos de cada instancia EC2 y el número de instancias que se han utilizado para obtenerlos.

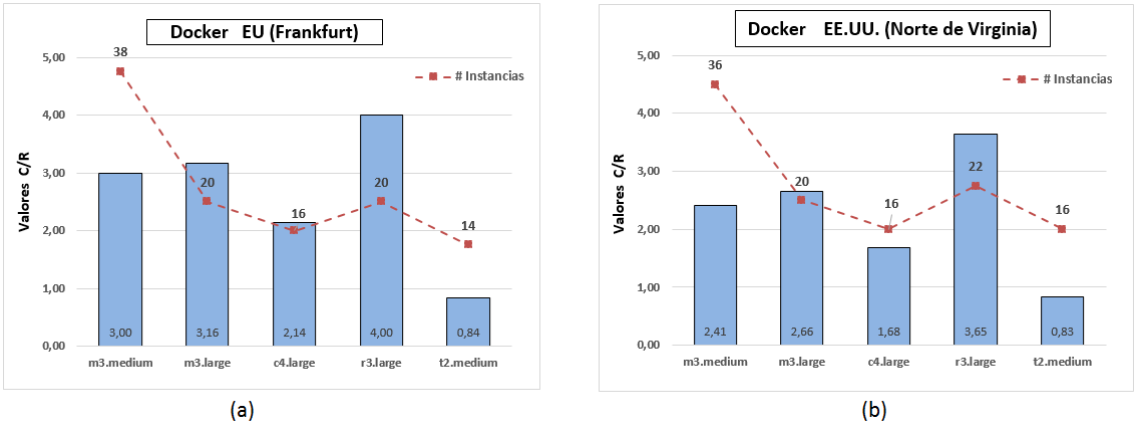

Figura 8.4. *Valores C/R al procesar 4 millones de ficheros, (a) en la Región EU (Frankfurt) y (b) en la Región EE.UU. (Norte de Virginia) de Amazon EC2 con Docker.*

Los tipos de instancias m3.medium en ambas regiones, utilizan un número de 20 instancias para obtener el valor C/R mínimo. En la región EU (Frankfurt) obtiene un valor C/R de 3,16 y en la región EE.UU. (Norte de Virginia) un valor C/R de 2,66.

Aun utilizando el mismo número de instancias y en una hora de ejecución, el factor que afecta en la diferencia de los valores C/R, son los precios que posee este tipo de instancia en cada una de las regiones.

La instancia t2.medium es la mejor opción para este caso, debido a que obtienen los mejores valores C/R mínimos, para la región EU (Frankfurt) un valor C/R de 0,84 y para la región EE.UU. (Norte de Virginia) un valor C/R de 0,83, siendo este último el mejor de ambas regiones, por tener el valor C/R mínimo.

Caso 3: Procesando 5 millones de ficheros y utilizando 20 instancias

En Amazon EC2:

Para analizar los datos de la región EU (Frankfurt), se deben analizar las Figuras 7.5 y 7.17, y para la región EE.UU. (Norte de Virginia), las Figuras 7.11 y 7.23.

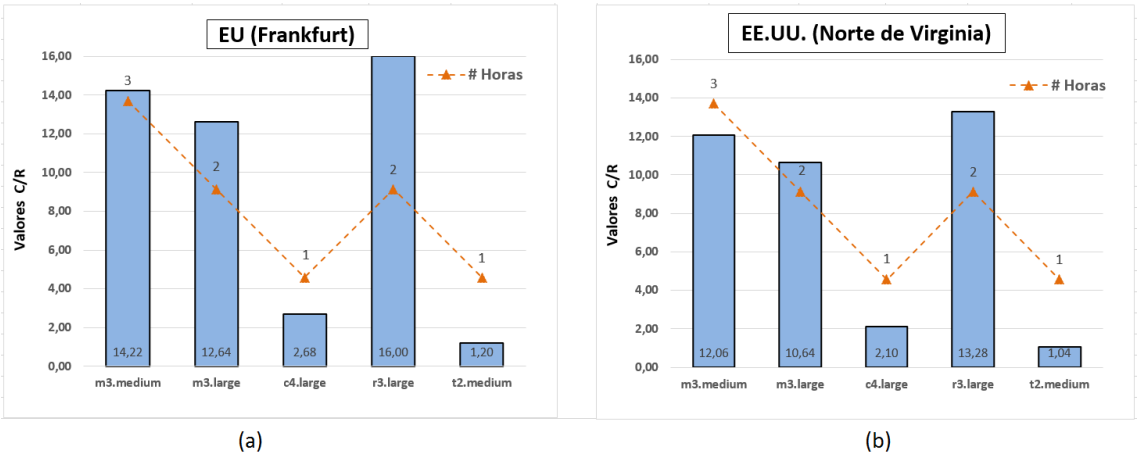

Figura 8.5. *Valores C/R al procesar 5 millones de ficheros y utilizando 20 instancias, (a) en la Región EU (Frankfurt) y (b) en la Región EE.UU. (Norte de Virginia) de Amazon EC2.*

Analizando los datos de las Figuras 8.5.a y 8.5.b, podemos observar que las instancias m3.large y r3.large, poseen similares características de cómputo y por ese motivo utilizan 2 horas de ejecución al utilizar 20 instancias, en ambas regiones. La diferencia en sus valores de C/R se debe a los precios que tienen, siendo la instancia m3.large la más económica. Por ese motivo, los valores C/R de la instancia m3.large son menores a los valores C/R de la instancia r3.large, en ambas regiones.

El tipo de instancia t2.medium es la mejor opción para este caso, debido a que obtienen los mejores valores C/R mínimos, para la región EU (Frankfurt) un valor C/R de 1,20 y para la región EE.UU. (Norte de Virginia) un valor C/R de 1,04, siendo esta ultima la mejor opción de ambas regiones.

En Amazon EC2 con Docker:

Para analizar los datos de la región EU (Frankfurt), se deben analizar las Figuras 7.29 y 7.41, y para la región EE.UU. (Norte de Virginia), las Figuras 7.35 y 7.47.

Observando las Figuras 8.6.a y 8.6.b, podemos observar que en este caso los tipos de instancias utilizan el mismo tiempo de ejecución en ambas regiones. Las diferencias de

los valor C/R, se deben a los precios que poseen, siendo la región EE.UU. (Norte de Virginia) la más económica, por ese motivo en esta región todos los tipos de instancias, obtienen los mejores valores C/R.

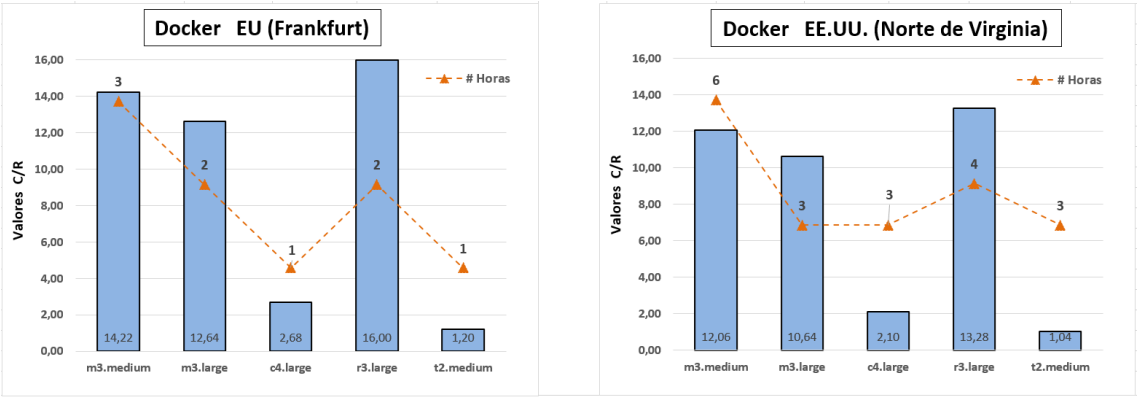

Figura 8.6. *Valores C/R al procesar 5 millones de ficheros y utilizando 20 instancias, (a) en la Región EU (Frankfurt) y (b) en la Región EE.UU. (Norte de Virginia) de Amazon EC2 con Docker.*

Podemos observar que los mejores valores C/R en ambas regiones, son obtenidos por los tipos de instancia c4.large y t2.medium. Siendo t2.medium la mejor opción para este caso, debido a que obtienen los mejores valores C/R mínimos, para la región EU (Frankfurt) un valor C/R de 1,20 y para la región EE.UU. (Norte de Virginia) un valor C/R de 1,04.

Con estos valores, podemos deducir que la instancia de la región EE.UU. (Norte de Virginia) es la mejor opción para este caso.

Caso 4: Procesando 6 millones de ficheros, utilizando 10 instancias y en un máximo de 3 horas

En Amazon EC2:

Para analizar los datos de la región EU (Frankfurt), se deben analizar las Figuras 7.6 y 7.18, y para la región EE.UU. (Norte de Virginia), las Figuras 7.12 y 7.24.

Observando la Figura 8.7.a, para este caso debemos descartar los tipos de instancias m3.medium, m3.large y r3.large, debido a que sobrepasan el tiempo de ejecución máximo de 3 horas. Entonces para este caso, solo pueden tomarse en cuenta los tipos de instancias t2.medium y c4.large, siendo t2.medium la que obtiene el valor C/R mínimo de 5,40.

Si observamos la Figura 8.7.b, el tipo de instancia m3.medium es descartado por tener un tiempo de ejecución de 6 horas, 3 horas más del tiempo máximo establecido para este

caso. Los demás tipos de instancias utilizan 3 horas de ejecución, siendo t2.medium la mejor instancia, debido a que obtiene un valor C/R de 4,68.

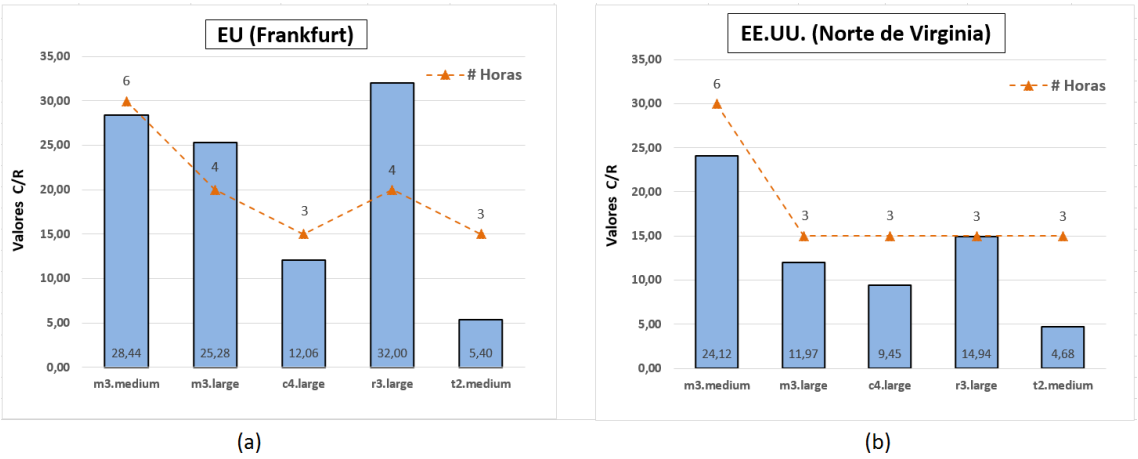

Figura 8.7. *Valores C/R al procesar 6 millones de ficheros, utilizando 10 instancias y en un máximo de 3 horas, (a) en la Región EU (Frankfurt) y (b) en la Región EE.UU. (Norte de Virginia) de Amazon EC2.*

Entonces se puede deducir que la instancia t2.medium de la región EE.UU. (Norte de Virginia), es el mejor tipo de instancia para este caso.

En Amazon EC2 con Docker:

Para analizar los datos de la región EU (Frankfurt), se deben analizar las Figuras 7.30 y 7.42, y para la región EE.UU. (Norte de Virginia), las Figuras 7.36 y 7.48.

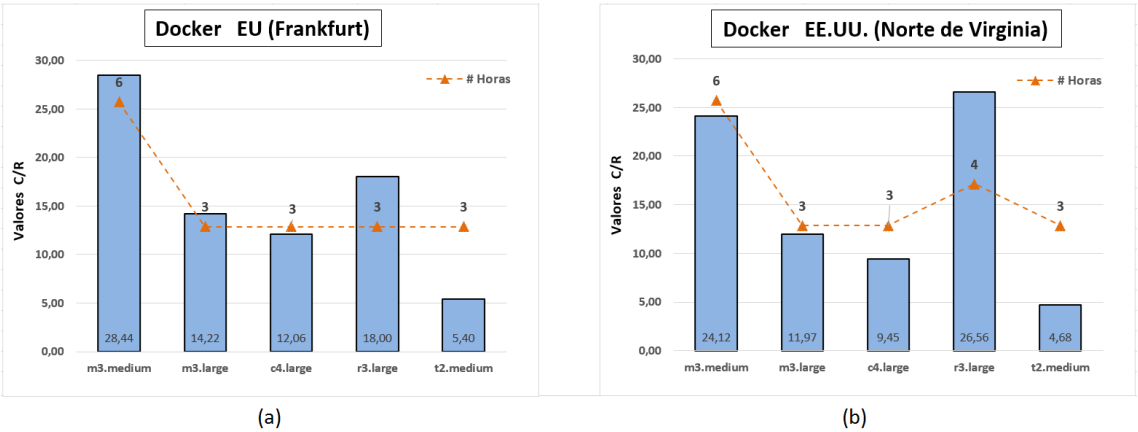

Figura 8.8. *Valores C/R al ejecutar 6 millones de ficheros, usando 10 instancias y en un máximo de 3 horas, (a) en la Región EU (Frankfurt) y (b) en la Región EE.UU. (Norte de Virginia) de Amazon EC2 con Docker.*

Observando la Figura 8.8.a, para este caso solo se debe descartar el tipo de instancia m3.medium, debido a que sobrepasa el tiempo de ejecución máximo de 3 horas. De las demás instancias, se seleccionó t2.medium como la mejor instancia al obtener un valor C/R de 5,40, el menor de todos los valores para la región EU (Frankfurt).

En región EE.UU. (Norte de Virginia) (ver Figura 8.8.b), solo se tendrán en cuenta los tipos de instancias m3.large, c4-large y t2.medium, debido a que no sobrepasan el límite máximo de 3 horas de ejecución. Siendo t2.medium la mejor instancia, debido a que obtiene un valor C/R de 4,68.

Entonces se puede deducir que la instancia t2.medium de la región EE.UU. (Norte de Virginia), es el mejor tipo de instancia para este caso.

9. Conclusiones y Trabajo futuro

El Cloud Computing nos proporciona los mejores niveles de servicio, una gran escalabilidad, infraestructuras mucho más fiables y una excelente disponibilidad de sus recursos en todo momento.

Después de haber desarrollado este trabajo, podemos mencionar que el Cloud Computing nos ofrece una solución para la ejecución de la aplicación ETL. Debido a que nos permite escalar hacia arriba o hacia debajo de una forma fácil y así evitar problemas al momento de procesar grandes cantidades de datos. Además, solo debemos pagar por el tiempo que usamos los recursos de computación, este modelo de negocio nos ayuda a reducir considerablemente los costes.

El Cloud Computing cuenta con una tecnología siempre actualizada y optimizada. Por ese motivo, los tiempos de ejecución obtenidos en Amazon EC2 fueron mejores que los obtenidos en la máquina local. Cada vez que la cantidad de datos a procesar se incrementaba, la diferencia de tiempos entre ambos ambientes aumentaba considerablemente.

La elección adecuada de los servicios de Amazon EC2, nos garantiza el mejor rendimiento de nuestra aplicación. Por ello, se cumplió con el objetivo de proporcionar la fórmula del modelo, que nos permite elegir la mejor configuración a partir del tiempo de ejecución, del coste y de los valores de Coste/Rendimiento.

Utilizando la fórmula del modelo, podemos mencionar que en los casos de Amazon EC2 y Amazon EC2 con Docker, el mejor tipo de instancia EC2 utilizado es la instancia t2.medium de la Región EE.UU. (Norte de Virginia).

Basándonos en las conclusiones antes mencionadas, se recomienda la migración de la aplicación ETL a la infraestructura Cloud de Amazon EC2.

Como posible trabajo futuro, en la fórmula del módulo se tendría en cuenta el número de cores que posee cada tipo de instancia EC2. Además, en caso de que se use Docker o no, se tomará en cuenta los tiempos de instalación y configuración de la aplicación ETL en las instancias EC2.

9. Conclusions and Future Work

Cloud Computing provides us with the best levels of service, scalability, much more reliable infrastructure and excellent availability of resources at all times.

Having developed this work, we can mention that Cloud Computing offers a solution for the execution of the ETL application. Because it allows us to scale up or under an easy way to avoid problems when processing large amounts of data. In addition, we only pay for the time we use computing resources, this business model helps us to significantly reduce costs.

Cloud Computing has a constantly updated and optimized technology. For this reason, the execution times obtained in Amazon EC2 were better than those obtained on the local machine. Whenever the amount of data to be processed is increased, the time difference between the two environments increased considerably.

The right choice of Amazon EC2 services, guarantees the best performance of our application. Therefore, it met the objective of providing the model formula, which allows us to choose the best configuration from the runtime, cost and Cost / Performance values.

Using the formula of the model, we can mention that in the case of Amazon EC2 and Amazon EC2 with Docker, the best EC2 instance used is the t2.medium instance of the US Region (Northern Virginia).

Based on the above findings, the migration of the ETL application to Amazon EC2 Cloud infrastructure is recommended.

As a possible future work, in the model formula would have the number of cores possessed by each EC2 instance type into account. In addition, if Docker is used or not, it will take into account the time of installation and configuration of the ETL application in EC2 instances.

Referencias Bibliográficas

[1] P. Kokkinos, T. A. Varvarigou, A. Kretsis, P. Soumplis, and E. A. Varvarigos, "Cost and Utilization Optimization of Amazon EC2 Instances," in *2013 IEEE Sixth International Conference on Cloud Computing*, 2013, pp. 518–525.

[2] "Amazon Web Services http://aws.amazon.com."

[3] "RackSpace www.rackspace.com."

[4] "Microsoft Azure www.azure.microsoft.com."

[5] Z. Ou, H. Zhuang, A. Lukyanenko, J. K. Nurminen, P. Hui, V. Mazalov, and A. Yla-Jaaski, "Is the Same Instance Type Created Equal? Exploiting Heterogeneity of Public Clouds," *IEEE Trans. Cloud Comput.*, vol. 1, no. 2, pp. 201–214, 2013.

[6] P. Mell and T. Grance, "The NIST Definition of Cloud Computing Recommendations of the National Institute of Standards and Technology."

[7] D. Kondo, B. Javadi, P. Malecot, F. Cappello, and D. P. Anderson, "Cost-benefit analysis of Cloud Computing versus desktop grids," in *2009 IEEE International Symposium on Parallel & Distributed Processing*, 2009, pp. 1–12.

[8] S. Ostermann, A. Iosup, N. Yigitbasi, R. Prodan, T. Fahringer, and D. Epema, "A Performance Analysis of EC2 Cloud Computing Services for Scientific Computing," Springer Berlin Heidelberg, 2010, pp. 115–131.

[9] M. Malathi, "Cloud computing concepts," in *2011 3rd International Conference on Electronics Computer Technology*, 2011, vol. 6, pp. 236–239.

[10] M. D. H. Parekh and D. R. Sridaran, "An Analysis of Security Challenges in Cloud Computing," *Int. J. Adv. Comput. Sci. Appl.*, vol. 4, no. 1, 2013.

[11] Y. Jadeja and K. Modi, "Cloud computing - concepts, architecture and challenges," in *2012 International Conference on Computing, Electronics and Electrical Technologies (ICCEET)*, 2012, pp. 877–880.

[12] Z. Hill and M. Humphrey, "A quantitative analysis of high performance

computing with Amazon's EC2 infrastructure: The death of the local cluster?," in *2009 10th IEEE/ACM International Conference on Grid Computing*, 2009, pp. 26–33.

[13] S. Narula, A. Jain, and Prachi, "Cloud Computing Security: Amazon Web Service," in *2015 Fifth International Conference on Advanced Computing & Communication Technologies*, 2015, pp. 501–505.

[14] H. Wang, Q. Jing, R. Chen, B. He, Z. Qian, and L. Zhou, "Distributed systems meet economics: pricing in the cloud," *Proceedings of the 2nd USENIX conference on Hot topics in cloud computing*. USENIX Association, pp. 6–6, 2010.

[15] F. Bracci, A. Corradi, and L. Foschini, "Database security management for healthcare SaaS in the Amazon AWS Cloud," in *2012 IEEE Symposium on Computers and Communications (ISCC)*, 2012, pp. 000812–000819.

[16] P. Barham, B. Dragovic, K. Fraser, S. Hand, T. Harris, A. Ho, R. Neugebauer, I. Pratt, A. Warfield, P. Barham, B. Dragovic, K. Fraser, S. Hand, T. Harris, A. Ho, R. Neugebauer, I. Pratt, and A. Warfield, "Xen and the art of virtualization," in *Proceedings of the nineteenth ACM symposium on Operating systems principles - SOSP '03*, 2003, vol. 37, no. 5, p. 164.

[17] G. Wang and T. S. E. Ng, "The Impact of Virtualization on Network Performance of Amazon EC2 Data Center," in *2010 Proceedings IEEE INFOCOM*, 2010, pp. 1–9.

[18] S. Ostermann, A. Iosup, N. Yigitbasi, R. Prodan, T. Fahringer, and D. Epema, "A Performance Analysis of EC2 Cloud Computing Services for Scientific Computing," Springer Berlin Heidelberg, 2010, pp. 115–131.

[19] J. Lobo, Y. Wang, F. Qi, and B. Yin, *Decision model for provisioning virtual resources in Amazon EC2*. [IFIP], 2012.

[20] "Amazon Web Services official website, http://aws.amazon.com/es/ec2/."

[21] G. Juve, E. Deelman, K. Vahi, G. Mehta, B. Berriman, B. P. Berman, and P. Maechling, "Data Sharing Options for Scientific Workflows on Amazon EC2," in *2010 ACM/IEEE International Conference for High Performance Computing,*

Networking, Storage and Analysis, 2010, pp. 1–9.

[22] T. Dörnemann, E. Juhnke, and B. Freisleben, "On-Demand Resource Provisioning for BPEL Workflows Using Amazon's Elastic Compute Cloud," in *2009 9th IEEE/ACM International Symposium on Cluster Computing and the Grid*, 2009, pp. 140–147.

[23] J. Dejun, G. Pierre, and C.-H. Chi, "EC2 Performance Analysis for Resource Provisioning of Service-Oriented Applications," Springer Berlin Heidelberg, 2010, pp. 197–207.

[24] "Amazon Web Services official website, http://aws.amazon.com/es/about-aws/global-infrastructure/."

[25] "Amazon Web Services official website, https://aws.amazon.com/es/ec2/instance-types/."

[26] "Docker project official website. https://www.docker.com/what-docker."

[27] B. Varghese, L. T. Subba, L. Thai, and A. Barker, "Container-Based Cloud Virtual Machine Benchmarking," in *2016 IEEE International Conference on Cloud Engineering (IC2E)*, 2016, pp. 192–201.

[28] C. Pahl and B. Lee, "Containers and Clusters for Edge Cloud Architectures -- A Technology Review," in *2015 3rd International Conference on Future Internet of Things and Cloud*, 2015, pp. 379–386.

[29] K. Seo, H. Hwang, I. Moon, O. Kwon, and B. Kim, "Performance comparison analysis of linux container and virtual machine for building cloud," 2014.

[30] C. Boettiger, "An introduction to Docker for reproducible research," *ACM SIGOPS Oper. Syst. Rev.*, vol. 49, no. 1, pp. 71–79, Jan. 2015.

[31] M. Amaral, J. Polo, D. Carrera, I. Mohomed, M. Unuvar, and M. Steinder, "Performance Evaluation of Microservices Architectures Using Containers," in *2015 IEEE 14th International Symposium on Network Computing and Applications*, 2015, pp. 27–34.

[32] "Docker Engine, https://docs.docker.com/engine/understanding-docker/."

[33] "Docker Hub - https://hub.docker.com/."

[34] “Docker project official website. https://www.docker.com/enterprise.”

[35] C. Zheng and D. Thain, “Integrating Containers into Workflows,” in *Proceedings of the 8th International Workshop on Virtualization Technologies in Distributed Computing - VTDC ’15*, 2015, pp. 31–38.

[36] “Docker Docs, https://docs.docker.com/.”

[37] N. Anand and M. Kumar, “Modeling and optimization of extraction-transformation-loading (ETL) processes in data warehouse: An overview,” in *2013 Fourth International Conference on Computing, Communications and Networking Technologies (ICCCNT)*, 2013, pp. 1–5.

[38] J. L. Vázquez-Poletti, G. Barderas, I. M. Llorente, and P. Romero, “A Model for Efficient Onboard Actualization of an Instrumental Cyclogram for the Mars MetNet Mission on a Public Cloud Infrastructure,” Springer Berlin Heidelberg, 2012, pp. 33–42.